Towards Adaptive Off-Policy Evaluation of Ranking Policies under Agnostic and Stochastic Behavior Models

Haruka Kiyohara[*]
Tokyo Institute of Technology
kiyohara.h.aa@m.titech.ac.jp

Nobuyuki Shimizu
Yahoo Japan Corporation
nobushim@yahoo-corp.jp

Yasuo Yamamoto
Yahoo Japan Corporation
yasyamam@yahoo-corp.jp

ABSTRACT

In many real-world recommender and search systems, presenting a ranked list of relevant items is crucial for increasing user engagement or revenue. *Off-Policy Evaluation* (OPE) of ranking policies is thus gaining growing attention, as it enables offline evaluation of new policies using only logged data. *Inverse Propensity Scoring* (IPS) is a prevalent approach in (general) OPE. Unfortunately, a naive application of IPS in the ranking setting often faces a critical variance issue due to the combinatorially large action space and the resulting huge importance weight. To reduce the variance, existing estimators introduce some user behavior assumptions to eliminate unnecessary importance weight. However, a strong assumption may in turn incur serious bias, making "assumption selection" a challenging problem. To tackle this problem, we propose the *Adaptive IPS* estimator, which interpolates among the existing estimators. AIPS does this by using a class of importance weights that include those of existing estimators. By tuning the interpolation hyperparameters of the importance weight in a data-driven way, the proposed estimator adaptively reduces the variance of IPS without incurring high bias. The empirical results demonstrate that the proposed estimator works reliably well across a range of user behavior models, including stochastic ones.

ACM Reference Format:
Haruka Kiyohara, Nobuyuki Shimizu, and Yasuo Yamamoto. 2022. Towards Adaptive Off-Policy Evaluation of Ranking Policies under Agnostic and Stochastic Behavior Models. In *KDD-UC '22: ACM SIGKDD COnference on Knowledge Discovery and Data Mining, Undergraaduate Consortium.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/xxx.xxx

1 INTRODUCTION

Interactive bandits and reinforcement learning (RL) policies are widely used in *ranking* systems, e.g., music streaming, search, and online advertising. While the *logging* or *behavior* policy presents relevant items to users, it also collects logged data valuable for evaluating and redesigning the ranking systems. For example, a music streaming system records the ranked list of songs it presented (i.e., playlist) and to which songs the user listened. This gives the system a chance to redesign the policy for a more relevant recommendation. Moreover, logged data are also beneficial for *Off-Policy Evaluation* (OPE) [14, 15], which aims to accurately evaluate the performance of *counterfactual* or *evaluation* policies using only offline logged data, without interacting with actual users. OPE is of great practical interest, as it can be a safe and costless substitute for online A/B tests [3, 6]. However, exploiting the logged data is challenging, as the logs are inherently biased due to the distribution shift between the behavior and evaluation policies.

A dominant approach to deal with the distribution shift in OPE is to use the importance sampling technique referred to as *Inverse Propensity Scoring* (IPS) [13, 18]. IPS enables an unbiased estimation of the policy performance, thus is widely used in (general) contextual bandit settings. However, a critical limitation of IPS is that its variance can be high when the action space is large [16]. Particularly in the *slate* contextual bandit setting where we present a ranked list of items (i.e., actions) to the users, IPS often struggles with extremely high variance due to the combinatorially large action space. To reduce the impractically large action space to a tractable one, existing estimators introduce some *user behavior assumptions.* For example, Li et al. [10] assumes that a user interacts with the presented actions independently at each position. Since this independence assumption confines the action-reward dependency within the same position, the resulting *Independent IPS* (IIPS) greatly reduces the variance of IPS by discarding the irrelevant importance weight. However, when the true user behavior is more complex, IIPS yields serious bias. In response to the bias issue of IIPS, McInerney et al. [11] proposed *Reward interaction IPS* (RIPS) based on the *cascade* assumption. The cascade assumption assumes that a user interacts with actions one-by-one from the top position [5]. RIPS is unbiased in more cases than IIPS, while reducing the variance of IPS. However, when the cascade assumption does not hold, RIPS still incurs serious bias in estimation, as we demonstrate in the experiment in Section 4. These bias-variance tradeoffs of existing estimators often make practical applications of OPE quite challenging, as the true user behavior model is usually unknown. Moreover, when the user behavior models are determined by chance for each user, a suitable estimator and assumption are difficult to identify.

Contributions. To safely and adaptively achieve a reasonable bias-variance tradeoff, this paper proposes a new OPE estimator that interpolates among the existing estimators. Our key idea in enabling the interpolation of the existing estimators is to leverage the nested structure among the importance weights of IPS, RIPS, and IIPS. Specifically, we define a class of importance weights based on the nested structure to include the importance weights of the existing estimators with some built-in hyperparameters. By tuning these hyperparameters in a data-driven manner, the resulting *Adaptive IPS* estimator is able to balance the bias-variance tradeoff without

[*] This work was done during HK's internship at Yahoo Japan Corporation.

KDD-UC '22, August 17, 2022, Washington, D.C.
© 2022 Association for Computing Machinery.
https://doi.org/10.1145/xxx.xxx

any prior knowledge about the (true) user behavior model. Finally, the empirical results demonstrate that AIPS is able to reduce the variance of IPS under relatively simple user behavior models (e.g., independent), while also avoiding high bias under complex and stochastic user behavior models.

Our contributions are summarized as follows.

- We propose AIPS, which adaptively interpolates among the existing estimators to balance both bias and variance.
- We empirically verify that the proposed estimator works stably well across various user behavior models, including stochastic ones.

2 PRELIMINARIES

This section describes the problem setup and summarizes the existing estimators and their statistical properties.

2.1 Setup

We consider a *slate* contextual bandit setting. Let $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ be a context vector (e.g., user demographics) and $\mathcal{A}$ be a finite set of discrete actions (e.g., songs). Let $\boldsymbol{a} = (a_1, a_2, \dots, a_l, \dots, a_L)$ be a slate action vector (e.g., a ranked list of songs) where L is the length of a slate (slate size). Following Kiyohara et al. [7], we call a function $\pi : \mathcal{X} \to \Delta(\mathcal{A}^L)$ a *factorizable* policy. Given context $\boldsymbol{x} \in \mathcal{X}$, it chooses an action at each slot (a_l) independently, where $\pi(\boldsymbol{a}|\boldsymbol{x}) := \prod_{l=1}^{L} \pi(a_l|\boldsymbol{x})$ is the probability of choosing a slate action vector $\boldsymbol{a}$. In contrast, we call $\pi : \mathcal{X} \to \Delta(\Pi_L(\mathcal{A}))$ a *non-factorizable* policy, where $\Pi_L(\mathcal{A})$ is a set of L-permutation of $\mathcal{A}$. Comparing two policies, a *factorizable* one may choose the same action more than twice in the slate, while a *non-factorizable* one chooses a slate action without any duplicates among slots (i.e., $\forall 1 \le k < l \le L, a_k \ne a_l$). Let $\boldsymbol{r} = (r_1, r_2, \dots, r_l, \dots, r_L)$ be a reward vector, where r_l is a random variable representing the **slot**-level reward observed at slot l (e.g., whether the recommended song at slot l results in a click). We consider the following weighted sum of slot-level rewards as an aggregated reward metric called **slate**-level reward [7, 11]:

$$r^* = \sum_{l=1}^{L} \alpha_l r_l,$$

where α_l denotes a non-negative weight for slot l. Note that, we use $q(\boldsymbol{x}, \boldsymbol{a}) := \mathbb{E}_{p(\boldsymbol{r}|\boldsymbol{x},\boldsymbol{a})}[r^*|\boldsymbol{x}, \boldsymbol{a}]$ to denote the **slate**-level mean reward function and $q_l(\boldsymbol{x}, \boldsymbol{a}) := \mathbb{E}_{p(\boldsymbol{r}|\boldsymbol{x},\boldsymbol{a})}[r_l|\boldsymbol{x}, \boldsymbol{a}]$ to denote the **slot**-level mean reward function.

Let $\mathcal{D} := \{(\boldsymbol{x}^{(i)}, \boldsymbol{a}^{(i)}, \boldsymbol{r}^{(i)})\}_{i=1}^{n}$ be logged bandit data with n independent observations. $\boldsymbol{a}^{(i)}$ is a vector of discrete variables indicating which slate action is chosen for individual i. $\boldsymbol{x}^{(i)}$ and $\boldsymbol{r}^{(i)}$ denote the context and reward vectors observed for i. We assume that the logged data are generated by a *behavior policy* π_0:

$$\{(\boldsymbol{x}^{(i)}, \boldsymbol{a}^{(i)}, \boldsymbol{r}^{(i)})\}_{i=1}^{n} \sim \prod_{i=1}^{n} p(\boldsymbol{x}^{(i)}) \pi_0(\boldsymbol{a}^{(i)}|\boldsymbol{x}^{(i)}) p(\boldsymbol{r}^{(i)}|\boldsymbol{x}^{(i)}, \boldsymbol{a}^{(i)}).$$

Throughout the paper, we assume that all confounders and slot-level rewards $(r_1, \dots, r_L)$ are observed, and the logged data have full support over slate actions (i.e., $\pi_0(\boldsymbol{a}|\boldsymbol{x}) > 0, \forall(\boldsymbol{x}, \boldsymbol{a})$).

For a function $f(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{r})$, we use

$$\mathbb{E}_n[f(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{r})] := n^{-1} \sum_{(\boldsymbol{x}^{(i)}, \boldsymbol{a}^{(i)}, \boldsymbol{r}^{(i)}) \in \mathcal{D}} f(\boldsymbol{x}^{(i)}, \boldsymbol{a}^{(i)}, \boldsymbol{r}^{(i)})$$

to denote its empirical expectation over n observations in $\mathcal{D}$. We also let $\mathbb{E}_{\mathcal{D}}[\cdot] := \mathbb{E}_{\prod_{i=1}^{n} p(\boldsymbol{x}^{(i)}) \pi_0(\boldsymbol{a}^{(i)}|\boldsymbol{x}^{(i)}) p(\boldsymbol{r}^{(i)}|\boldsymbol{x}^{(i)}, \boldsymbol{a}^{(i)})}[\cdot]$. Regarding the action and reward vectors $(\boldsymbol{a}, \boldsymbol{r})$, we use the following notations.

- partial set of slate actions: $\boldsymbol{a}_{l_1:l_2} := (a_{l_1}, a_{l_1+1}, \dots, a_{l_2-1}, a_{l_2})$
- partial set of slate rewards: $\boldsymbol{r}_{l_1:l_2} := (r_{l_1}, r_{l_1+1}, \dots, r_{l_2-1}, r_{l_2})$

2.2 Estimation Target

We are interested in estimating the following *policy value* of any given *evaluation policy* π using only logged data $\mathcal{D}$:

$$V(\pi) := \mathbb{E}_{p(\boldsymbol{x})\pi(\boldsymbol{a}|\boldsymbol{x})p(\boldsymbol{r}|\boldsymbol{x},\boldsymbol{a})}[r^*].$$

Estimating the policy value before deploying π in an online environment is beneficial, as it does not require huge implementation costs of A/B tests [14] and mitigates the risks of damaging user satisfaction [3]. However, deriving an accurate estimation is also challenging due to the *distribution shift* between the evaluation (π) and behavior (π_0) policies.

2.3 Existing Estimators

Here, we review the related estimators, their user behavior assumptions, and their statistical properties.

Inverse Propensity Scoring. IPS uses the importance sampling technique to deal with the distribution shift as follows.

$$\hat{V}_{\mathrm{IPS}}(\pi; \mathcal{D}) := \mathbb{E}_n \left[\frac{\pi(\boldsymbol{a}|\boldsymbol{x})}{\pi_0(\boldsymbol{a}|\boldsymbol{x})} \sum_{l=1}^{L} \alpha_l r_l \right],$$

When the policy is factorizable, IPS is also described as follows.

$$\hat{V}_{\mathrm{IPS}}(\pi; \mathcal{D}) = \mathbb{E}_n \left[\left(\prod_{l=1}^{L} \frac{\pi(a_l \mid \boldsymbol{x})}{\pi_0(a_l \mid \boldsymbol{x})} \right) \sum_{l=1}^{L} \alpha_l r_l \right].$$

IPS is unbiased and consistent without any user behavior assumptions. However, it suffers from impractically high variance when the action space ($|\mathcal{A}^L|$ or $|\Pi_L(\mathcal{A})|$) is large [7, 10, 11, 19].

Independent Inverse Propensity Scoring. IIPS assumes the independence assumption, where a user interacts with actions independently across slots [7, 10].[1] This means that the reward observation at each slot depends only on the corresponding action and its position, but not the other actions presented in the same slate. Under this assumption, the slot-level mean reward function is reduced to the following.

$$q_l(\boldsymbol{x}, \boldsymbol{a}) = \mathbb{E}_{p(r_l|\boldsymbol{x},a_l)}[r_l \mid \boldsymbol{x}, a_l].$$

Based on the above condition, IIPS corrects the distribution shift only at the corresponding position (l) for each reward (r_l) as follows.

$$\hat{V}_{\mathrm{IIPS}}(\pi; \mathcal{D}) := \mathbb{E}_n \left[\sum_{l=1}^{L} \frac{\pi(a_l \mid \boldsymbol{x})}{\pi_0(a_l \mid \boldsymbol{x})} \alpha_l r_l \right],$$

[1] The independence assumption corresponds to the item-position model of [10].

where $\pi(a_l \mid x) := \sum_{\boldsymbol{a}'} \pi(\boldsymbol{a}|\boldsymbol{x})\,\mathbb{I}(a'_l = a_l)$ is the action choice probability at slot l. IIPS drastically reduces the variance of IPS, while remaining unbiased under the independence assumption. However, when this assumption does not hold, IIPS incurs serious bias in estimation [7, 11].

Reward interaction Inverse Propensity Scoring. RIPS assumes the cascade assumption, i.e., a user interacts with actions sequentially from the top position to the bottom [5]. Therefore, the reward observed at each slot (r_l) is dependent only on actions and rewards at higher positions in a ranking ($\boldsymbol{a}_{l+1:L}, \boldsymbol{r}_{l+1:L}$). Since r_l is independent of the lower positions ($l+1 : L$), the slot-level mean reward function results in the following.

$$q_l(\boldsymbol{x}, \boldsymbol{a}) = \mathbb{E}_{p(r_l|\boldsymbol{x},\boldsymbol{a}_{1:l},\boldsymbol{r}_{1:l-1})}\left[r_l \mid \boldsymbol{x}, \boldsymbol{a}_{1:l}, \boldsymbol{r}_{1:l-1}\right].$$

Note that, the cascade assumption includes the independence assumption as a special case.

Exploiting the cascade assumption, RIPS estimates the policy value as follows.

$$\begin{aligned}\hat{V}_{\mathrm{RIPS}}(\pi; \mathcal{D}) &:= \mathbb{E}_n\left[\sum_{l=1}^{L} \frac{\pi(\boldsymbol{a}_{1:l} \mid \boldsymbol{x})}{\pi_0(\boldsymbol{a}_{1:l} \mid \boldsymbol{x})}\alpha_l r_l\right] \\ &= \mathbb{E}_n\left[\sum_{l=1}^{L}\left(\prod_{l'=1}^{l} \frac{\pi_e(a_{l'} \mid \boldsymbol{x}, \boldsymbol{a}_{1:l'-1})}{\pi_b(a_{l'} \mid \boldsymbol{x}, \boldsymbol{a}_{1:l'-1})}\right)\alpha_l r_l\right].\end{aligned}$$

RIPS is unbiased under the cascade assumption, while also reducing the variance of IPS. However, when the cascade assumption does not hold, its bias becomes high, as we will empirically verify in the experiment in Section 4.

2.4 Limitation of Existing Estimators under Agnostic and Stochastic Behavior Models

So far, we have seen that the existing estimators exhibit different bias-variance tradeoffs depending on their user behavior assumptions. When the (true) user behavior model is deterministic and known a priori, practitioners can pick the most suitable estimator based on the user behavior assumption. However, the true user behavior often follows an unknown probability distribution – for example, users may follow either the cascade assumption or the independence assumption with probability 0.8 and 0.2 in platform A, while users do not follow any assumptions in platform B. In such cases, applying low-variance estimators such as IIPS and RIPS can be risky, as they may aggravate the estimation error when the assumption does not hold. In fact, the empirical results of Kiyohara et al. [7] indicate that a suitable estimator and assumption can change depending on the (true) user behavior model in the data generation process. This motivates us towards an *adaptive* OPE estimator that can interpolate among IPS, RIPS, and IIPS to better balance the bias-variance tradeoff in a data-driven way.

3 ADAPTIVE OFF-POLICY EVALUATION

In this section, we propose a new OPE estimator called *Adaptive Inverse Propensity Scoring* (AIPS).

Table 1: Correspondence among the interpolation hyperparameters of AIPS and the existing estimators

estimator	assumption	λ	λ^-	λ^+
IPS	none	1	1	1
RIPS	cascade	1	1	0
IIPS	independence	1	0	0

Proposed Method. Our key insight in deriving a new estimator is that the action choice probability of an arbitrary policy can be expressed by the following nested structure.

$$\pi(\boldsymbol{a}|\boldsymbol{x}) = \underbrace{\pi(a_l|\boldsymbol{x})\pi(\boldsymbol{a}_{1:l-1}|\boldsymbol{x}, a_l)}_{\pi(\boldsymbol{a}_{1:l}|\boldsymbol{x})}\pi(\boldsymbol{a}_{l+1:L}|\boldsymbol{x}, \boldsymbol{a}_{1:l}). \tag{1}$$

Leveraging this structure, AIPS is able to interpolate among IPS, RIPS, and IIPS with a class of importance weights, which include the importance weights of the existing estimators as follows.

$$\begin{aligned}&\hat{V}_{\mathrm{AIPS}}(\pi; \mathcal{D}) \\ &:= \mathbb{E}_n\left[\sum_{l=1}^{L} w_l(\boldsymbol{x}, \boldsymbol{a}; \lambda) w_{1:l-1}(\boldsymbol{x}, \boldsymbol{a}; \lambda^-) w_{l+1:L}(\boldsymbol{x}, \boldsymbol{a}; \lambda^+)\,\alpha_l r_l\right],\end{aligned}$$

where we define each importance weight as[2]:

$$\begin{aligned}w_l(\boldsymbol{x}, \boldsymbol{a}; \lambda) &:= \lambda\,\frac{\pi(a_l|\boldsymbol{x})}{\pi_0(a_l|\boldsymbol{x})} + (1 - \lambda) \\ &\quad\left(= \frac{\lambda\,\pi(a_l|\boldsymbol{x}) + (1 - \lambda)\pi_0(a_l|\boldsymbol{x})}{\pi_0(a_l|\boldsymbol{x})}\right) \\ w_{1:l-1}(\boldsymbol{x}, \boldsymbol{a}; \lambda^-) &:= \lambda^- \frac{\pi(\boldsymbol{a}_{1:l-1}|\boldsymbol{x}, a_l)}{\pi_0(\boldsymbol{a}_{1:l-1}|\boldsymbol{x}, a_l)} + (1 - \lambda^-) \\ w_{l+1:L}(\boldsymbol{x}, \boldsymbol{a}; \lambda^+) &:= \lambda^+ \frac{\pi(\boldsymbol{a}_{l+1:L}|\boldsymbol{x}, \boldsymbol{a}_{1:l})}{\pi_0(\boldsymbol{a}_{l+1:L}|\boldsymbol{x}, \boldsymbol{a}_{1:l})} + (1 - \lambda^+)\end{aligned}$$

where $1 \geq \lambda \geq \lambda^- \geq \lambda^+ \geq 0$ are the interpolation hyperparameters. Table 1 shows the correspondence among AIPS and the existing estimators. When $\lambda^* = 1$, AIPS preserves the original importance weight to correct the distribution shift. On the other hand, when $\lambda^* = 0$, AIPS eliminates the importance weight and ignores the distribution shift for the variance reduction purpose. More generally, when the value of λ^* is large, AIPS leads to a low-bias but high-variance estimator, while a small value of λ^* leads to a high-bias but low-variance estimator.

Hyperparameter Tuning. To adaptively balance the bias-variance tradeoff, we minimize the following estimated MSE.

$$\begin{aligned}&\hat{\mathrm{MSE}}(\hat{V}_*(\cdot, \lambda^*)) \\ &:= \left(\mathbb{E}_{\mathcal{D}^b}\left[\hat{V}_*(\cdot, \lambda^*))\right] - \mathbb{E}_{\mathcal{D}^b}\left[\hat{V}_{\mathrm{IPS}}(\cdot)\right]\right)^2 + \eta\hat{\mathbb{V}}(\hat{V}_*(\cdot, \lambda^*))\end{aligned}$$

where η is a hyperparameter. $\mathbb{E}_{\mathcal{D}^b}[\cdot]$ denotes the bootstrapped mean. The first and second terms are the estimated bias and variance, respectively. We estimate the bias by substituting the true policy value $V(\cdot)$ with $\mathbb{E}_{\mathcal{D}^b}[\hat{V}_{\mathrm{IPS}}(\cdot)]$, since V is unobservable and $\hat{V}_{\mathrm{IPS}}$ is unbiased under any user behavior model. However, as $\hat{V}_{\mathrm{IPS}}$ is vulnerable to variance, this procedure tends to overestimate bias. Therefore, we use η to balance the bias-variance tradeoff. We compare several choices of η in the following experiment.

[2] These importance weights coincide with the *arithmetic* correction of [12].

Table 2: Estimators' mean-squared-error (MSE), squared bias, and variance under a single user behavior model (i.e., $\gamma = 1.0$)

	standard			cascade			independent		
OPE Estimators	MSE	squared bias	variance	MSE	squared bias	variance	MSE	squared bias	variance
IPS	**0.400***	0.004	0.396	0.368	0.004	0.364	0.365	0.004	0.361
RIPS	**1.804**†	1.598	0.206	**0.160***	0.004	0.156	**0.150**$^\diamond$	0.004	0.146
IIPS	**6.943**†	6.822	0.121	**1.474**†	1.435	0.039	**0.019***	0.006	0.013
AIPS ($\eta = 1.0$)	0.461	0.005	0.455	**0.332**$^\diamond$	0.006	0.326	**0.172**$^\diamond$	0.006	0.166
AIPS ($\eta = 2.0$)	0.514	0.037	0.477	**0.347**$^\diamond$	0.018	0.329	**0.120**$^\diamond$	0.007	0.113
AIPS ($\eta = 3.0$)	0.568	0.084	0.484	**0.363**$^\diamond$	0.033	0.329	**0.092**$^\diamond$	0.009	0.084
AIPS (oracle)	**0.183***	0.085	0.098	**0.245**$^\diamond$	0.002	0.243	**0.217**$^\diamond$	0.017	0.200

Note: A lower value is better for all metrics. The **red*** fonts represent the most accurate OPE estimator. The **green**$^\diamond$ fonts represents the OPE estimators that improve MSE upon IPS. The **blue**† fonts represent the OPE estimators that aggravate the MSE of IPS more than twice.

4 SYNTHETIC EXPERIMENT

This section compares AIPS with the existing estimators.[3] Our synthetic experiment uses *OpenBanditPipeline* [14][4]. The omitted experimental details are provided in Appendix.

4.1 Setup

Basic Setting. We collect the size $n = 1,000$ of the logged data using a factorizable policy. It chooses each slot-level action independently as $a_l \sim \pi(a_l|\boldsymbol{x})$, where we set $|\mathcal{A}| = 3$.[5] We also set $L = 6$ and $\alpha_l = 1, \forall l = 1, \dots, L$. The slot-level reward is continuous, which is sampled from a normal distribution as $r_l \sim \mathcal{N}(q_l(\boldsymbol{x}, \boldsymbol{a}), \sigma^2)$, where $q_l(\boldsymbol{x}, \boldsymbol{a})$ and σ are the slot-level mean reward function and the noise level of the rewards, respectively. The following describes how $q_l(\boldsymbol{x}, \boldsymbol{a})$ is defined in various user behavior models.

User behavior models and rewards. We use *standard*, *cascade*, and *independent* user behavior models, following Kiyohara et al. [7]. To model each user behavior, we first introduce the following general form of the slot-level mean reward function:

$$q_l(\boldsymbol{x}, \boldsymbol{a}) := \tilde{q}_l(\boldsymbol{x}, a_l) + F(\boldsymbol{x}, \boldsymbol{a}),$$

where $\tilde{q}_l(\boldsymbol{x}, a_l)$ is the base reward function, which is determined only by a_l. In contrast, $F(\boldsymbol{x}, \boldsymbol{a})$ is dependent on the whole slate action $\boldsymbol{a}$. $F(\boldsymbol{x}, \boldsymbol{a})$ models the different action-reward dependencies of the user behavior models as follows.

$$F(\boldsymbol{x}, \boldsymbol{a}) = \begin{cases} \sum_{k \neq l} G(k, l) & \text{(standard)} \\ \sum_{k < l} G(k, l) & \text{(cascade)} \\ 0 & \text{(independent)} \end{cases},$$

where $G(k, l)$ is the effect of the action presented at slot k on the slot l. Specifically, when the user behavior model is standard, q_l depends on the whole slate ($\boldsymbol{a}$). In contrast, q_l depends only on the higher slots ($\boldsymbol{a}_{1:l}$) under the cascade behavior model, while q_l is dependent only on the corresponding slot (a_l) under the independent one. To see how the population of each user behavior model affects the performance of each OPE estimator, we let γ of data sampled from the *target* user behavior model, and $(1 - \gamma)$ of data sampled evenly from the other two. For example, when $\gamma = 0.4$ and the target is standard, 40% of the data is based on standard, 30% is on cascade, and 30% is on independent. We vary $\gamma \in \{0.0, 0.2, \dots, 1.0\}$ for each target user behavior model.

Compared estimators. We compare IPS, IIPS, RIPS, and AIPS (our proposal). Note that, IPS is unbiased under all user behavior models. RIPS is unbiased under the cascade and independent, while IIPS is unbiased only under the independent one. We use the true action choice probability of π_b to define the compared estimators.

For AIPS, we select each λ^* from $\Lambda := \{0.0, 0.1, \dots, 1.0\}$ in the order of $\lambda^+, \lambda^-, \lambda$.[6] We bootstrap data 100 times to derive the bootstrapped mean. As discussed, the hyperparameter tuning method tends to overestimate bias due to the variance of $\hat{V}_{\text{IPS}}(\cdot)$. Therefore, we compare AIPS with $\eta \in \{1.0, 2.0, 3.0\}$. We also compare AIPS (oracle), which is tuned with the bias estimated by the true policy value (i.e., $V(\cdot)$).

4.2 Results and Discussions

We conduct the experiment with 1000 different seeds and calculate the mean-squared-error (MSE) as the estimators' performance metric. We also decompose MSE into bias and variance for analysis.

How estimators perform differently under the single user behavior setting? We first validate the performance of the compared estimators under a single user behavior model (i.e., $\gamma = 1.0$). Table 2 summarizes the estimators' performance metrics under each user behavior model. The result suggests that the existing low-variance estimators (i.e., RIPS and IIPS) enable the most accurate estimation when the user behavior assumption holds. However, when the assumption does not hold, those estimators produce extremely high bias, aggravating the MSE of the baseline estimator (i.e., IPS) more than twice. In contrast, AIPS do not incur such high bias, ensuring a reliable estimation even under the most complex user behavior model (i.e., standard). Moreover, when the true user behavior model is less complex (in particular in the independent case), AIPS successfully reduces the variance of IPS. These favorable performances are due to the adaptive interpolation of AIPS.

[3] The code is provided at: https://github.com/aiueola/kdd-uc-2022-adaptive-ips.

[4] https://github.com/st-tech/zr-obp

[5] Since we are using a factorizable policy as π_b, the behavior policy is able to choose the same action more than twice in a slate.

[6] The order is based on the inner-slate action dependency described in Eq. (1).

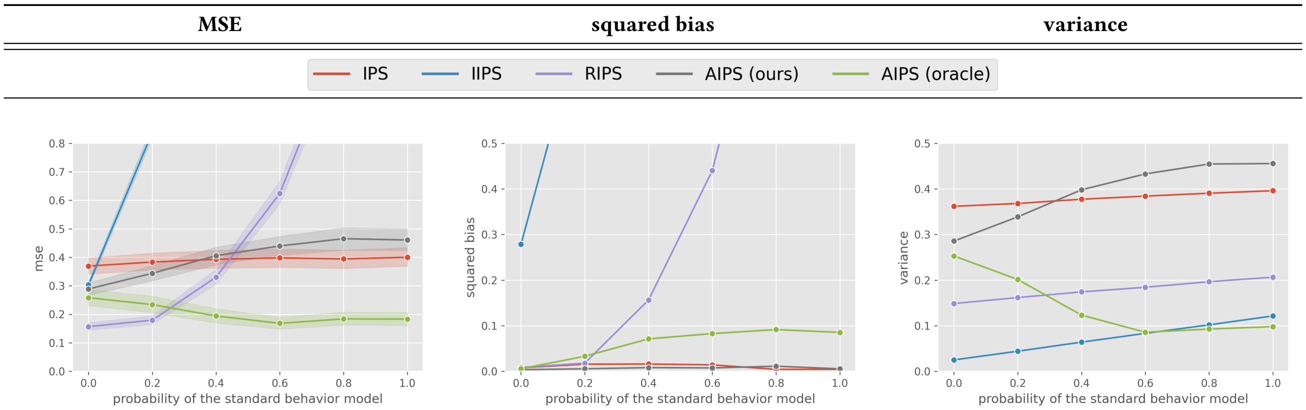

Figure 1: Estimators' performance comparison with the varying probabilities (γ) of the standard behavior model

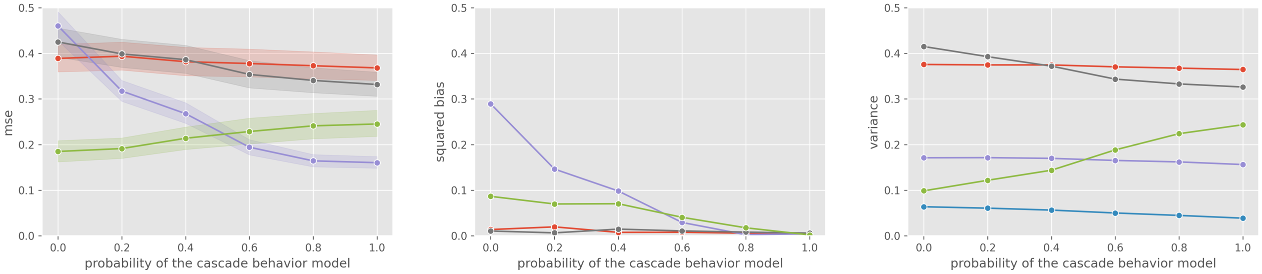

Figure 2: Estimators' performance comparison with the varying probabilities (γ) of the cascade behavior model

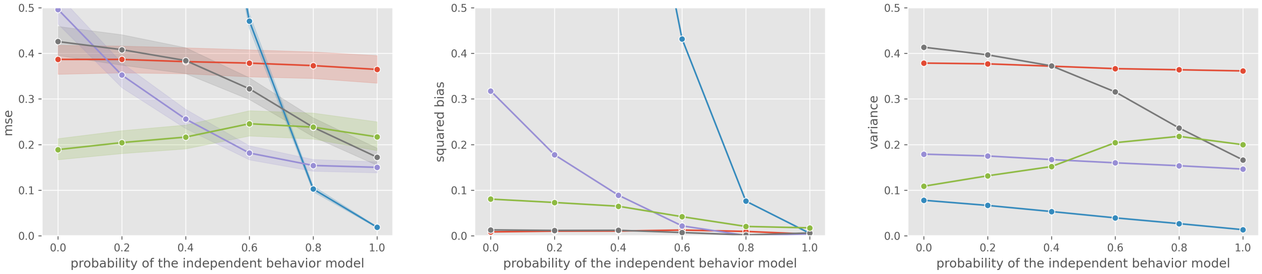

Figure 3: Estimators' performance comparison with the varying probabilities (γ) of the independent behavior model

Note: The shaded regions in MSE plots show the bootstrapped 95% confidence intervals. We report the result of $\eta = 1.0$ for AIPS. Note that, in Figure 2, IIPS is out of range in the plots of MSE and squared bias due to high bias.

However, the results also indicate that AIPS has room for improvement with respect to hyperparameter tuning. First, we observe that the variance reduction of AIPS is smaller than that of RIPS and IIPS. In particular, the variance of AIPS is more than twice of that of RIPS in the cascade case, suggesting that AIPS is too conservative in interpolating low-variance estimators. We hypothesize that this observation is due to the overestimation of bias in the tuning procedure. To verify this hypothesis, we also compare AIPS with the varying values of η. We observe that a large value of η reduces variance when the true user behavior is simple (i.e., independent). However, the result also indicates that a large value of η can increase both bias and variance when the true user behavior is relatively complex, as a large value of η makes AIPS too myopic in variance reduction. These observation suggests that hyperparameter tuning is challenging due to the uncertainty in the bias estimation. Finally, the oracle tuning result (i.e., AIPS (oracle)) suggests that AIPS potentially reduces MSE by a larger margin than the existing estimators when hyperparameter tuning works successfully. We leave the development of reliable hyperparameter tuning methods for future work.

How estimators perform when the user behavior models are stochastic? Next, we investigate the situation where a population of users follows complex user behavior models, but another population of users follows simple user behavior models, following some unknown probability distribution. This situation is more practically relevant than a single user behavior case, but is under-explored in the existing OPE literature in the ranking setting. Figure 1-3 illustrate how the estimators' performance metrics change with the varying probabilities (γ) of the *target* user behavior models. Remarkably, the result demonstrates that the estimation error of the low-variance but biased estimators increases more than proportional as the probability of the complex user behavior models increases. This is because the probability of the simple user behavior models also decreases at the same time as the increase of the complex user behavior models. Therefore, assuming a user behavior assumption might be risky in the stochastic user behavior case, as it potentially leads to an extremely inaccurate OPE estimation. In contrast, we observe that AIPS is able to reduce the variance of IPS when the independent user behavior model occupies a certain probability (i.e., $\gamma \geq 0.6$), without incurring high bias under the complex user behavior models (i.e., standard and cascade). We highlight that our proposed estimator's smooth interpolation among the existing estimators enables a reliably accurate estimation across various situations, making the practical application of OPE more tractable.

5 RELATED WORK

OPE is of great practical relevance in recommender systems, as it enables the performance evaluation of counterfactual policies using only logged data, without interacting with users in the field [6, 8, 15, 17]. Especially, the single item (action) recommendation setting referred to as (general) contextual bandit has extensively been studied [3, 4, 9, 16]. Direct Method (DM) [1], IPS [13, 18], and Doubly Robust (DR) [2] are the three prevalent methods. DM uses a machine learning model to estimate the mean reward function ($q(\boldsymbol{x}, a)$). Then, it takes the expectation of the estimated reward ($\hat{q}(\boldsymbol{x}, a)$) over the evaluation policy. Although DM is reasonably accurate when $\hat{q}$ is accurate, DM is vulnerable to the bias caused by model-misspecification [2]. In contrast, IPS is a model-free approach, which exploits the importance sampling technique to correct the distribution shift. IPS is unbiased, but can suffer from high variance [2]. DR is a hybrid of DM and IPS, which uses the predicted reward as a control variate and performs importance weighting only on the residual of the estimation. DR usually reduces the variance of IPS while remaining unbiased. However, DR can still incur high variance when the action space is large [16].

In the *slate* contextual bandit setting where we present a ranking consisting of several items (actions) to users, OPE faces the challenges of a combinatorially large action space. Specifically, naive applications of IPS often face extremely high variance [7, 10, 11, 19]. To tackle the variance, existing work has introduced some user behavior assumptions to reduce the combinatorial action space to a tractable one. In particular, IIPS [10] is based on the independence assumption, which assumes that a user interacts with actions independently across slots. Under this assumption, the reward observed at each slot depends only on the corresponding action at the same position. Therefore, IIPS is able to ignore the difference in actions presented in the other slots, leading to a significant variance reduction compared to IPS. While IIPS is unbiased when the independence assumption holds, however, this strong assumption generally does not hold in real world data, resulting in a serious bias [7, 11]. RIPS [11] balances both bias and variance by assuming the cascade assumption, i.e., a user interacts with actions one-by-one from the top position [5]. Therefore, the reward observed at each slot depends only on the actions presented at higher positions, excluding the action-reward interactions from the lower positions. RIPS is unbiased under the cascade assumption, which includes the independence assumption as a special case, while also reducing the variance of IPS. However, RIPS still incurs bias in estimation when the cascade assumption does not hold. In addition, RIPS suffers from a high variance when the slate size is large [7]. To tackle the latter variance problem, Kiyohara et al. [7] propose Cascade-DR, leveraging the recursive structure of the cascade assumption. However, the former bias problem has not been addressed in the existing literature. Therefore, we are the first to work on the adaptive "assumption selection" problem. In particular, our proposed estimator adaptively interpolates among the existing estimators, which are based on the different user behavior assumptions. We highlight that the smooth interpolation of AIPS enables a reliably accurate estimation even under the stochastic behavior models, which is practically relevant but under-explored in the existing literature.

Finally, PI [19, 20] is another OPE estimator in the slate contextual bandit setting. This estimator considers a situation where the slot-level rewards (r_l) are unobservable and only the slate-level reward (r^*) is observed. Therefore, PI is not suitable when the slot-level rewards are observable, as it discards the information of slot-level rewards. Moreover, PI is also prone to have a serious bias due to the independence assumption, as empirically verified in McInerney et al. [11].

6 CONCLUSION AND FUTURE WORK

This paper studied OPE of ranking policies in the slate contextual bandit setting. In this setting, the existing estimators have relied on a single and pre-defined user behavior assumption to improve the variance upon IPS. However, when the assumption does not hold, the existing low-variance estimators may exacerbate the estimation accuracy due to catastrophically high bias. To reduce the variance of IPS without incurring high bias, we developed the assumption-free estimator called *Adaptive IPS* based on the interpolated importance weight. Since AIPS adaptively interpolates among the existing estimators in a data-driven manner, AIPS is able to achieve a moderate bias-variance tradeoff agnostic to the user behavior models. Finally, the empirical results indicate that AIPS works stably accurate on various user behavior models, including the stochastic ones, even when the other low-variance estimators struggle with high bias.

In future work, we plan to explore a reliable hyperparameter tuning method for OPE of ranking policies. As discussed, the hyperparameter tuning is quite challenging in this setting, as the bias estimation often suffers from high variance. However, we believe that a reliable hyperparameter tuning procedure would greatly improve the safety and applicability of OPE of ranking policies in practical situations.

REFERENCES

[1] Alina Beygelzimer and John Langford. 2009. The Offset Tree for Learning with Partial Labels. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 129–138.

[2] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning* (Bellevue, Washington, USA) *(ICML'11)*. Omnipress, Madison, WI, USA, 1097–1104.

[3] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. 2018. Offline A/B Testing for Recommender Systems. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 198–206.

[4] Alois Gruson, Praveen Chandar, Christophe Charbuillet, James McInerney, Samantha Hansen, Damien Tardieu, and Ben Carterette. 2019. Offline Evaluation to Make Decisions About Playlist Recommendation Algorithms. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 420–428.

[5] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient Multiple-Click Models in Web Search. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*. 124–131.

[6] Haruka Kiyohara, Kosuke Kawakami, and Yuta Saito. 2021. Accelerating Offline Reinforcement Learning Application in Real-Time Bidding and Recommendation: Potential Use of Simulation. *arXiv preprint arXiv:2109.08331* (2021).

[7] Haruka Kiyohara, Yuta Saito, Tatsuya Matsuhiro, Yusuke Narita, Nobuyuki Shimizu, and Yasuo Yamamoto. 2022. Doubly Robust Off-Policy Evaluation for Ranking Policies under the Cascade Behavior Model. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. 487–497.

[8] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv preprint arXiv:2005.01643* (2020).

[9] 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. *WSDM*, 297–306.

[10] Shuai Li, Yasin Abbasi-Yadkori, Branislav Kveton, S Muthukrishnan, Vishwa Vinay, and Zheng Wen. 2018. Offline Evaluation of Ranking Policies with Click Models. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1685–1694.

[11] James McInerney, Brian Brost, Praveen Chandar, Rishabh Mehrotra, and Benjamin Carterette. 2020. Counterfactual Evaluation of Slate Recommendations with Sequential Reward Interactions. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1779–1788.

[12] Alberto Maria Metelli, Alessio Russo, and Marcello Restelli. 2021. Subgaussian and Differentiable Importance Sampling for Off-Policy Evaluation and Learning. 34 (2021).

[13] Doina Precup, Richard S. Sutton, and Satinder P. Singh. 2000. Eligibility Traces for Off-Policy Policy Evaluation. In *Proceedings of the 17th International Conference on Machine Learning*. 759–766.

[14] Yuta Saito, Shunsuke Aihara, Megumi Matsutani, and Yusuke Narita. 2020. Open Bandit Dataset and Pipeline: Towards Realistic and Reproducible Off-Policy Evaluation. *arXiv preprint arXiv:2008.07146* (2020).

[15] Yuta Saito and Thorsten Joachims. 2021. Counterfactual Learning and Evaluation for Recommender Systems: Foundations, Implementations, and Recent Advances. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 828–830.

[16] Yuta Saito and Thorsten Joachims. 2022. Off-Policy Evaluation for Large Action Spaces via Embeddings. *arXiv preprint arXiv:2202.06317* (2022).

[17] Yuta Saito, Takuma Udagawa, Haruka Kiyohara, Kazuki Mogi, Yusuke Narita, and Kei Tateno. 2021. Evaluating the Robustness of Off-Policy Evaluation. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 114–123.

[18] Alex Strehl, John Langford, Lihong Li, and Sham M Kakade. 2010. Learning from Logged Implicit Exploration Data. In *Advances in Neural Information Processing Systems*, Vol. 23. 2217–2225.

[19] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. 2017. Off-Policy Evaluation for Slate Recommendation. In *Advances in Neural Information Processing Systems*, Vol. 30. 3632–3642.

[20] Nikos Vlassis, Fernando Amat Gil, and Ashok Chandrashekar. 2021. Off-Policy Evaluation of Slate Policies under Bayes Risk. *arXiv preprint arXiv:2101.02553* (2021).

A OMITTED EXPERIMENTAL DETAILS

Here, we provide some additional experimental setups omitted in the main text.

Contexts and Rewards. To generate synthetic data, we first sample five-dimensional contexts (i.e., $d = 5$), independently and normally distributed with zero mean. Then, based on the context $\boldsymbol{x}$ and the action presented at each slot a_l, the base reward function $\tilde{q}_l(\boldsymbol{x}, a_l)$ is defined as follows.

$$\tilde{q}_l(\boldsymbol{x}, a_l) := \left[\theta_{a_l}^{\top}\boldsymbol{x} + b_{a_l}\right]_+,$$

where we let $[z]_+ := \max\{z, 0\}$. θ_{a_l} is a parameter vector sampled from the standard normal distribution. b_{a_l} is a bias term that corresponds to action a_l.

On the other hand, we use the *additive* reward function [7] to model the interaction among slots as $G(k, l) = \mathbb{W}(a_k, a_l)$. $\mathbb{W}$ is $|\mathcal{A}| \times |\mathcal{A}|$ symmetric matrix which defines how an action affects the reward of the other actions in the same slate. This additive interaction simulates the effect of co-occurrence between two actions.

Finally, we set the reward noise as $\sigma = 5.0$ when sampling the reward from the mean reward function (q_l).

Behavior and evaluation policies. We use the following factorizable policy as the behavior policy.

$$\pi_b(\boldsymbol{x}, \boldsymbol{a}) = \prod_{l=1}^{L} \pi_b(\boldsymbol{x}, a_l) = \prod_{l=1}^{L} \mathrm{softmax}\left(f_b(\boldsymbol{x}, a_l)\right),$$

where $f_b(\boldsymbol{x}, a_l) = \theta_{a_l}^{\top}\boldsymbol{x}+b_{a_l}$. The parameters θ_{a_l} and b_{a_l} are sampled from the standard uniform distribution.

Then, we define the evaluation policy based on the (pre-defined) behavior policy as follows.

$$\pi_e(\boldsymbol{x}, \boldsymbol{a}) = \prod_{l=1}^{L} \mathrm{softmax}\left(\lambda \cdot f_b(\boldsymbol{x}, a_l) + (1 - |\lambda|)\right),$$

where $\lambda \in [-1.0, 1.0)$ is a hyperparameter that controls the distribution shift between π_b and π_e. A positive value of λ leads to an evaluation policy that is similar to the behavior policy (i.e., small distribution shift). On the other hand, a negative λ leads to an evaluation policy that deviates from the behavior policy greatly. We set $\lambda = -5.0$ in our experiment.