Counterfactual Evaluation of Slate Recommendations with Sequential Reward Interactions

James McInerney[1*], Brian Brost[2*], Praveen Chandar[2*]
Rishabh Mehrotra[3], Ben Carterette[2]
jmcinerney@netflix.com;brianbrost,praveenr,rishabhm,benjaminc@spotify.com
[1] Netflix, Los Gatos, CA [2] Spotify, New York, NY
[3] Spotify, London, UK

ABSTRACT

Users of music streaming, video streaming, news recommendation, and e-commerce services often engage with content in a sequential manner. Providing and evaluating good sequences of recommendations is therefore a central problem for these services. Prior reweighting-based counterfactual evaluation methods either suffer from high variance or make strong independence assumptions about rewards. We propose a new counterfactual estimator that allows for sequential interactions in the rewards with lower variance in an asymptotically unbiased manner. Our method uses graphical assumptions about the causal relationships of the slate to reweight the rewards in the logging policy in a way that approximates the expected sum of rewards under the target policy. Extensive experiments in simulation and on a live recommender system show that our approach outperforms existing methods in terms of bias and data efficiency for the sequential track recommendations problem.

1 INTRODUCTION

Recommender systems enable users of online services to navigate vast libraries of content (e.g. news, music, videos). Evaluating such systems is a central challenge for these services, with A/B testing often regarded as the gold standard [7, 12]. In A/B testing users are randomly assigned to different recommendation algorithms to isolate the treatment effects between them. Unfortunately, A/B tests are costly in several ways: significant effort is required to implement new recommenders into production, the tests take weeks or months to run, and revenue is lost from sub-optimal user experiences if the existing recommender turns out to be better. Against

* Equal contribution authors.

this background, offline evaluation is crucial in rapidly validating new recommenders to ensure that only the strongest ideas are tested online.

Offline evaluation is challenging because the deployed recommender decides which items the user sees, introducing significant exposure bias in logged data [6, 15, 21]. Various methods have been proposed to mitigate bias using counterfactual evaluation. In this paper, we use terminology from the multi-armed bandit framework to discuss these methods: the recommender performs an *action* by showing an item depending on the observed *context* (e.g., user covariates, item covariates, time of day, day of the week) and then observes a *reward* through the user response (e.g., a stream, a purchase, or length of consumption) [13]. The recommender follows a *policy* distribution over actions by drawing items stochastically conditioned on the context.

The basic idea of counterfactual evaluation is to estimate how a new policy would have performed if it had been deployed instead of the deployed policy. This is challenging because it is easy to be fooled by spurious correlations between the reward and context in logged data. For example, logged data may indicate that pop music recommendations have a higher average reward but this effect may not hold up in intervention (i.e., through a new recommendation) if the production policy had been systematically recommending pop to younger users who may be more active on the service regardless of genre preferences.

The prevailing method for counterfactual evaluation is inverse propensity scoring (IPS). IPS approximates the average reward of a target policy offline by taking a weighted average of the rewards obtained under the production policy. The weights are a function of the ratio between target and production policies evaluated on the logged data. IPS estimators are usually unbiased (or asymptotically unbiased) but suffer high variance with large action spaces, a problem that is typical in recommendation. While simplifying assumptions can be made to reduce variance, they in turn introduce bias, particularly when there are interactions between rewards. When recommendations are sequential, as in a playlist of tracks, these interactions sometimes dominate the overall reward. For example, a good song in an early position of a sequence can have a large effect on skip rates of songs at subsequent positions.

A major challenge in rapidly iterating on sequential recommendation systems is a reliable offline evaluation methodology. Sequential recommendations are becoming increasingly relevant due to richer interactions between users and recommenders [8, 16, 26, 28]. In such scenarios, the user's action (or reward) on a recommended item depends on the recent recommendations or user's previous actions. For example, a user purchasing a mobile phone is more

likely to buy accessories in the near term. These effects are often not captured by traditional recommendation systems.

In this paper, we introduce *reward interaction inverse propensity scoring* (RIPS), a new approach to IPS estimation on sequences of sub-actions that avoids the modeling overhead of model-based estimation (i.e., the *direct method*) and reduces the variance through structural assumptions about the reward dependencies in a slate. Our approach assumes a causal graph of conditional independencies of actions and rewards, enabling us to derive a counterfactual estimator that does not require distributional assumptions or parameter fitting.

In summary, our contributions are the following:

- We formulate a new off-policy estimator of the total rewards of a slate in counterfactual evaluation based on causal graph assumptions about the interactions between the context, actions, and rewards in the slate. The new estimator is a nested expectation describing how reweighting factors accumulate down the slate.
- To approximate the nested expectation, we propose a tractable algorithm [1] that uses iterative normalization and lookback to estimate the average reward of the target policy from finite data collected using a logging policy.
- Experiments show that the improved properties of RIPS enable it to recover, with significantly greater accuracy than prior state-of-the-art, the ground truth value of the target policy both in simulation and the outcome of A/B tests in a live sequential recommendation system.

The rest of the paper is organized as follows. In Section 2, we review related work in the area of counterfactual evaluation. In Section 3, we present the RIPS algorithm. Sections 4 and 5 describe the set of experiments in simulation and online (respectively) to compare RIPS against other baselines for counterfactual evaluation. Finally, we conclude and discuss future work in Section 6.

2 RELATED WORK

In this section, we discuss prior work on evaluating recommender system using implicit and explicit feedback both online and offline.

2.1 Online evaluation

The most trusted method for measuring the quality of a recommendation system, conventional or sequential, is using online controlled experiments. In this setup, comparing two or more recommendation algorithms involves redirecting the user traffic to different variants [12] and measuring the quality of recommendation algorithm, often based on a predefined user satisfaction metric. In this work, we rely on a simple A/B test setup to estimate the true rewards of different policies in our experiments.

2.2 Offline evaluation

While online evaluation is a reliable method of evaluation, it takes several weeks to collect sufficient amount of data to reliably compare systems. Further, there is the risk of user attrition since we could expose potentially bad recommendations to real users. On the other hand, offline evaluation is cheap and enables us to test ideas rapidly without exposing experimental systems to live traffic.

2.2.1 Traditional Methods. For several decades, recommender and search systems evaluations have heavily relied on offline test collections for rapid experimentation [20]; however, they are expensive and require a considerable amount of time and effort to curate. Furthermore, such collections cannot easily be adapted to account for personal or time-sensitive relevance. Alternatively, the use of explicit feedback such as user ratings has been explored, but prior research has pointed out that explicit feedback information collected is often missing information, not at random (MNAR) [17, 21, 22] leading to bias in evaluation.

Implicit signals such as clicks are a rich source of information for offline evaluation, since they are collected in a natural setting, typically reflecting the user's preferences and available at a low cost. However, recent work has pointed out that the implicit data collected is subject to different sources of biases [21]. Specifically, the feedback collected leads to sampling bias since the data collection is mediated by the recommendation system themselves; our focus is on addressing these biases in this work.

2.2.2 Counterfactual Methods. Counterfactual analysis techniques are increasingly used for training and evaluating machine learned recommendation models from user log data collected online. One specific technique is *inverse propensity scoring* (IPS) [18], which has a long history in the study of experimental design and clinical trials. IPS is a way to estimate the average reward of a target policy from data collected according to a logging policy while correcting for the mismatch between target and logging policy.

IPS estimators can be proven to be unbiased under weak assumptions. But even when those assumptions are true, IPS estimators suffer high variance with large action spaces [6]. Large action spaces occur when there are many items in the catalog and when recommendation happens in slates. Slate recommendation is an increasingly important and common interface design that presents a set of items to a user at once (e.g., search results pages, a shelf of videos, a playlist of tracks) and the user may engage with any or none of the items or *sub-actions* in the slate [25]. The sub-actions together affect the observed reward, making it difficult to disentangle the role of any particular sub-action. There are a combinatorially large number of potential actions resulting in a very small overlap between the production policy and target policy. Consequently, the weights in IPS are mostly zero, ignoring most of the data and giving a high variance estimate of the target policy reward.

Variance Reduction Methods: Capping IPS weights and normalizing them against their sum has been shown to reduce variance, but increase bias [6]. In other cases, simplifying assumptions can be deployed to reduce the variance of IPS estimators in slate recommendation. Li et al. assume that the reward for each sub-action is independent of other sub-actions in the slate [14]; this is a typical assumption to greatly reduce the size of the action space that is used in practical applications [4]. To simplify slate reward estimation, Swaminathan et al. take an additive approach to the slate reward and assume that the sub-action rewards are unobserved and independent [25]. In contrast, our method assumes that the sub-action rewards are observed and we do not assume independence. Looking at other problem settings, different schemes for lookback capping

---

[1]The code for the proposed estimator is available at https://github.com/spotify-research/RIPS_KDD2020

have been proposed to reduce variance for off-policy estimation in reinforcement learning [27] and to reduce bias in the non-stationary contextual bandit setting [10].

Model-Based or Direct methods: In general, with the exception of naïve IPS, which suffers from excessive variance, the bias introduced by existing IPS methods is large when the rewards for sub-actions in a slate are not independent. The only existing general approach to taking advantage of sequential structure in the rewards of sub-actions in a slate is the direct method [5]. In the direct method, a model of rewards is posited then trained on logged data. The direct method requires a significant amount of modeling overhead to specify the distribution of rewards, fitting their parameters from data, criticizing the model, and finding the right hyperparameters. This effort does not translate across datasets/applications and is vulnerable to overfitting. Doubly robust methods are well known to reduce the variance of IPS estimators [5]. but carry all the overhead of the direct method, and, due to the combinatorial explosion in the number of possible slates, even a good control variate will not alleviate the aforementioned issues with IPS. Related methods that combine multiple sub-estimators such as [27], and generalizations thereof [23] can improve the bias-variance tradeoff, but do not eliminate the aforementioned modeling overhead. An interesting future exploration could be to characterize how our estimator can be incorporated into these combined estimator families.

3 REWARD INTERACTION INVERSE PROPENSITY SCORING (RIPS)

In this section, we describe the problem of slate recommendation and discuss the shortcomings of inverse propensity scoring and its variants. Using causal assumptions about the actions and rewards in the slate, we present our algorithm for counterfactual evaluation.

3.1 Problem Formulation

Slate recommendation posits a sequence of sub-actions $A_{1:K}$ chosen by a policy π in a context X that results in a total reward R for the slate. In the scope of this work, we consider the setting where there is an observed reward R_k for each ordered sub-action A_k in the slate and the total reward is the sum of the sub-action rewards, i.e., $R = \sum_{k=1}^{K} R_k$.[2] We do not assume that the rewards are independent.

Suppose π is a recommender that takes contextual features about a user (e.g. device, time of day, long-term user features) and returns an ordered set of items. The overall reward for the slate is the total number of engagements (e.g. purchases, clicks, completions) and these engagements, though correlated, are associated with individual actions. An example of reward correlation occurs in playlist listening: if a user hears a song they dislike in a playlist, they may skip it and may then be more likely to skip the next song. In other settings, e.g., homepage recommendation, there is a negative correlation between the user response to items high in the list and those lower in the list due to a limited attention budget.

When run in production, π produces slates and logs impressions. We use superscript n ($1 \leq n \leq N$) to refer to the realization of the

random variable on impression n, e.g., $R_k^{(n)}$ is the reward for the kth action $A_k^{(n)}$ of the nth impressed slate $A^{(n)}$.

After running π in production, the feedback logs will consist of all the sampled contexts, actions, and rewards $\left\{ \left(X^{(n)}, A_{1:K}^{(n)}, R_{1:K}^{(n)} \right)_{n=1}^{N} \right\}$. In general, the size of the slate K may be different across impressions and there may be a context for every position in the slate. For simplicity of notation, we use constant K and $X^{(n)}$ across impressions, though this assumption is not required by our method.

The goal of counterfactual evaluation is to estimate the average reward for a target policy h. All methods in counterfactual evaluation estimate $V(h) = \mathbb{E}[\sum_{k=1}^{K} R_k]$ using the logged data. Our starting point in addressing this problem is a popular class of evaluators based on importance sample reweighting.

3.2 Inverse Propensity Scoring (IPS)

Inverse propensity scoring (IPS) methods use the importance sample reweighting method to evaluate a target policy h by reweighting the rewards received from collection policy π,

$$\hat{V}_{\text{IPS}}(h) = \frac{1}{N} \sum_{n=1}^{N} \frac{\prod_{k=1}^{K} h(A_k^{(n)} \mid X^{(n)})}{\prod_{k=1}^{K} \pi(A_k^{(n)} \mid X^{(n)})} \sum_{k=1}^{K} R_k^{(n)} \qquad (1)$$

where $X^{(n)} \sim p(X)$, $A^{(n)} \sim \pi(A \mid X^{(n)})$, $R^{(n)} \sim p(R \mid X^{(n)}, A^{(n)})$, and $p(X)$ and $p(R \mid X^{(n)}, A^{(n)})$ are the unknown distributions for the context and rewards.[3]

In this work, we examine IPS-based methods and their properties using the causal graph representation of Pearl's *do*-calculus [19]. A causal graph expresses conditional independencies between random variables in the same way a probabilistic graphical model does. In addition, the $do(X = x)$ operator on a random variable X represents an intervention to make $X = x$ that removes all incoming edges in the graph and forces downstream variables to condition on $X = x$. This approach allows us to formulate novel IPS methods to take advantage of sequential structure in the slate rewards.

IPS usually requires the stable unit treatment value assumption (SUTVA) in order to derive an estimator that reweights individual actions (i.e., units) [9]. The individual actions in slate recommendation violate SUTVA because the action selected in one position affects the outcome of actions in other positions. The typical strategy to address SUTVA violations is to change how actions are defined. Specifically, one can redefine actions in slate recommendation as the ordered sequence of sub-actions. The issue with doing so is that the action space increases combinatorially resulting in high variance with IPS. For this reason, standard IPS is limited in its applicability to slate estimation.

3.2.1 Independent IPS. In order to widen the applicability of IPS-methods to larger slates with different h and π, existing work uses an item-position click model that applies IPS under the assumption that the probability of a click depends only on the item and its

---

[2]Our derivations are trivial to extend to the more general case that the total reward for a slate is an affine function of the sub-action rewards. We focus on the sum of sub-action rewards for simplicity of exposition.

[3]N.B. all IPS-based methods discussed in this paper admit policies that condition on previous actions. This is possible because IPS interacts with known functions π and h via a score for each sub-action that may depend on the context and any previous sub-actions in the slate.

(a) Standard IPS

(b) Independent IPS and PI

(c) Reward interaction IPS (RIPS)

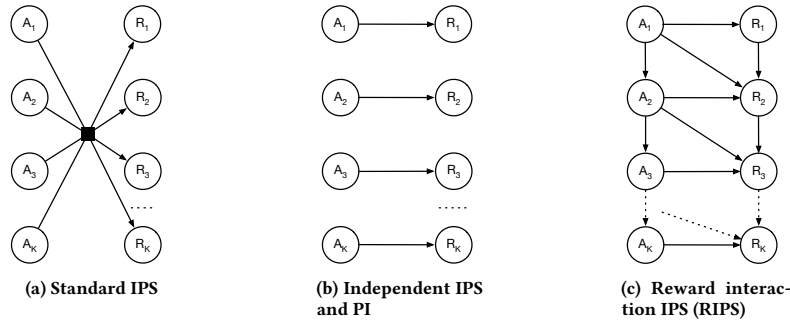Figure 1: The assumptions of each IPS-based estimator described in this work can be visualized as a causal graphical model.

position [14],

$$V_{\text{IIPS}}(h) = \mathbb{E}_X[\mathbb{E}_{A \sim h(\cdot \mid X), R \sim p(\cdot \mid X, A)}[\sum_{k=1}^{K} R_k]]$$

$$= \{\text{assume } R_k \text{ depends only on } X, \ k, \text{ and } A_k\}$$

$$\mathbb{E}_{X, A \sim h(\cdot \mid X)}[\sum_{k=1}^{K} \mathbb{E}_{R_k \sim p(\cdot \mid X, A_k)}[R_k]]$$

$$= \mathbb{E}_{X, A \sim \pi(\cdot \mid X)}[\sum_{k=1}^{K} \mathbb{E}_{R_k \sim p(\cdot \mid X, A_k)}[\frac{h(A_k \mid X)}{\pi(A_k \mid X)} R_k]]$$

$$\approx \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} w_k^{(n)} R_k^{(n)}, \quad \text{where } w_k^{(n)} = \frac{h(A_k^{(n)} \mid X^{(n)})}{\pi(A_k^{(n)} \mid X^{(n)})} \quad (2)$$

We refer to this estimator as *independent IPS* (IIPS) because the rewards are assumed independent of any other actions or rewards in the slate. IIPS is a more general version of the first-order approximation of IPS that treats positions identically [1, 4]. We focus on IIPS here because we found empirically that it has sufficiently low variance. We show the independence assumptions for IPS and IIPS graphically in Figures 1a and 1b. Nodes indicate random variables and arrows represent direct dependencies between random variables.

While IIPS has proven effective at counterfactual slate reward estimation, it assumes that each reward in the slate is isolated from every other action and reward. This rules out any sequential interactions between actions and rewards in the data. If this does not hold, the estimate will be biased no matter how much data is collected. Furthermore, the estimator becomes increasingly confident in the biased estimate with more data.

3.2.2 Pseudoinverse Estimator. The pseudoinverse estimator (PI) for off-policy evaluation [25] assumes that the observed slate-level reward is a known linear function of the sub-action rewards, and that the sub-action rewards are not observed. These, along with the absolute continuity assumption (i.e. that $h(A \mid X) > 0 \implies \pi(A \mid X) > 0, \ \forall A, X$), allow the off-policy evaluation problem to be restated as a linear regression problem, where the sub-actions and their positions are encoded in a feature vector $\mathbb{I}_A$ of ones and zeros (depending on slate A) and a set of unknown weights that

correspond to the sub-action rewards,

$$\hat{V}_{\text{PI}}(h) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_h[\mathbb{I}_{A^{(n)}} \mid X^{(n)}]^\top \mathbb{E}_\pi[\mathbb{I}_{A^{(n)}} \mathbb{I}_{A^{(n)}}^\top \mid X^{(n)}]^\dagger \mathbb{I}_{A^{(n)}} \sum_{k=1}^{K} R_k^{(n)},$$

where superscript † indicates the pseudoinverse of a matrix.

PI is not well suited to situations where the sub-action rewards are observable, i.e., when the credit for the success of a slate is attributable to a particular sub-action. Further, the linearity assumption means that the rewards for sub-actions are assumed independent. To loosen the independence assumptions of both IIPS and PI to produce a reliable estimate, we next introduce an extension to IPS that deals with sequences of actions and rewards in a slate.

3.3 Reward interaction IPS (RIPS)

Reward interaction IPS (RIPS) assumes a Markov structure to the rewards as shown in Fig. 1c. The reward at position k is directly conditional on the action performed at k, and the previous action and reward at position $k - 1$. The dependence of rewards on each other makes it very challenging to estimate the expected off-policy reward because the usual trick of replacing the expectation with an empirical average is not available in this case. Furthermore, the downstream effects of an intervention at position k could extend beyond $k + 1$ through cascading effects. To motivate the assumption underlying RIPS, we note that for the real-world music streaming data introduced in Section 5, the average skip rate is 0.41, but the skip rate given a prior skip is 0.78, and the skip rate given the previous track was not skipped is 0.21. This kind of sequential dependence is not taken into account by IIPS and PI.

Note that, although we specified conditional independence assumptions, we have not posited a full reward model that describes the effect of actions on rewards. (If we were to do this, we would be using the direct method.)[4] The advantage of IPS methods is that they avoid the additional burden of fitting parameters and dealing with the bias that action-reward model assumptions introduce.

---

[4]For example, specifying that the reward R_k is a parametric function of A_k and A_{k-1} and using this predictor to directly calculate the value of another policy.

In the first instance, we can apply the conditional independence assumptions to refine the IPS estimate into a nested expectation,

$$V_{\text{Nested}}(h) = \mathbb{E}_X[\mathbb{E}_{A\sim h,R}[\sum_{k=1}^{K} R_k]]$$

$$= \{\text{assume } R_k \text{ depends only on } X, \ k, \ A_k, \text{ and } R_{k-1}\}$$

$$\mathbb{E}_X[\mathbb{E}_{A_1\sim h,R_1}[R_1 + \mathbb{E}_{A_2\sim h,R_2}[R_2 + \dots$$

$$+ \mathbb{E}_{A_K\sim h,R_K}[R_K \mid R_{K-1}] \mid R_{K-2}]\dots \mid R_1]]], \quad (3)$$

where we suppressed some notation for readability. Applying importance sample reweighting results in nested reweightings with the same structure, under the usual assumption of absolute continuity,

$$V_{\text{Nested}}(h) = \mathbb{E}_X[\mathbb{E}_{A_1\sim\pi,R_1}[\frac{h(A_1 \mid X)}{\pi(A_1 \mid X)}(R_1 + \mathbb{E}_{A_2\sim\pi,R_2}[\frac{h(A_2 \mid X, A_1)}{\pi(A_2 \mid X, A_1)}$$

$$(R_2 + \quad \dots + \mathbb{E}_{A_K\sim\pi,R_K}[\frac{h(A_K \mid X, A_{K-1})}{\pi(A_K \mid X, A_{K-1})}R_K \mid R_{K-1}]) \mid R_{K-2}])$$

$$\dots \mid R_1])]]. \quad (4)$$

Eq. 4 indicates that extreme propensity scores are still a problem, as the jth term will have a weight of $\prod_{k=1}^{j} \frac{h(A_k \mid X, A_{k-1})}{\pi(A_k \mid X, A_{k-1})}$. To address this, we make use of normalization and *lookback capping*.

In more detail, the expected reweighting factor for any subsequence of k actions in the slate is 1 due to,

$$\mathbb{E}\left[\frac{h(A_{1:k})}{\pi(A_{1:k})}\right] = \int \frac{h(A_{1:k})}{\pi(A_{1:k})}\pi(A_{1:k})dA_{1:k} = \int h(A_{1:k})dA_{1:k} = 1. \quad (5)$$

This leads to an iterative normalization procedure for approximating the nested expectation in Eq. 3 based on accumulating a message γ_k through the reward chain such that γ_k is the reweighting factor for the reward at position k:

$$\hat{V}_{\text{RIPS}}(h) := \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma_k^{(n)}R_k^{(n)}, \quad (6)$$

$$\text{where } \gamma_k^{(n)} := \frac{N\gamma_{k-1}^{(n)}w_k^{(n)}}{\sum_{n'=1}^{N}\gamma_{k-1}^{(n')}w_k^{(n')}} \text{ and } \gamma_0^{(n)} := 1. \quad (7)$$

The reweighting factors for actions toward the end of the slate are still a product of a large number of weights (that are often close to zero) and normalization addresses but does not solve the problem. To further mitigate extreme weights, we propose a form of *lookback capping* to limit the effect of combining more weights than the data support. Lookback capping uses the effective sample size (ESS) [11] to estimate the amount of data that are available to provide an estimate at position k based on the overlap between the logging and target policy,

$$ESS(\gamma_k^{(1:N)}) = \frac{N^2}{\sum_{n=1}^{N}(\gamma_k^{(n)})^2} \quad (8)$$

Intuitively, as we increase the lookback at position k, the weights will become increasingly extreme, causing the effective sample size to fall below some threshold t that represents a proportion of the size of the dataset. We choose t to balance the number of interactions considered in the estimator against the high variance

introduced by extreme weights. The lookback should be as high as possible while also ensuring ESS > Nt.

The reason why it is not necessary to look forward when computing reweighting factors is because the outcome at position k is only affected by *do*-calculus interventions from position 1 to position k. This means that, in contrast to sequential inference in non-causal graphs, we only need to do filtering and no smoothing.

The full RIPS algorithm is presented in Algorithm 1 in Appendix A. It has time complexity $O(NK^2)$, where K is the slate size (usually a small constant). A test on line 12 of the algorithm is added to ensure that the ESS is always decreasing as b_k increases. In practice, the ESS increases after a large enough lookback because the more weights that are included, the greater the chance that every sample contains an extreme weight, causing normalization to weight the samples equally. This raises the ESS despite the fact that every sample is now equally bad.

When the conditional independence assumptions shown in Figure 1c hold, RIPS has lower variance than standard IPS (and its normalized variant NIS [6]) and is both consistent and asymptotically unbiased. More details about the properties of RIPS are provided in Appendix A.

4 SIMULATION EXPERIMENTS

We adopt a dual approach to empirically evaluate the performance of our proposed reward interaction inverse propensity scoring estimator. First, we use simulation experiments to illustrate how existing off-policy estimators are biased, or suffer from extreme variance, when the user's rewards are dependent on previous rewards. We also demonstrate the ability of our proposed estimator to deal with such reward interactions. In Section 5, we show the practical applicability of our problem setting and approach by testing how the estimators evaluate different recommendation policies on user data collected during an A/B test.

4.1 Simulation Setup

In this section, we describe the simulation setup we use to illustrate how simplifying assumptions made by existing off-policy estimators lead to biased estimates. Specifically, we show that in the presence of interactions between rewards, an off-policy estimator must explicitly account for the interactions to provide unbiased estimates. We designed experiments around simulations of contexts, slates, and user interactions on a given slate; this allowed us to analyze the counterfactual methods under a flexible and controlled environment. We briefly describe the components of our simulation setup below and report the results of various state-of-the-art estimators.

4.1.1 Context and reward simulation: We generate simulated contexts, each consisting of a set of candidate sub-actions. In this study, we use ten sub-actions for a given context. This setting reflects a typical sequential recommendation scenario. For example, consider the task of recommending tracks within an algorithmic playlist in online music streaming services where the context includes details of about user and the sub-actions are the tracks to be recommended. Next, we randomly assign the true reward (i.e., probability of streaming a track in the case of track recommendation) for the sub-actions in the candidate pool for each context. Note that the true rewards are not known to the off-policy estimators in the simulation.
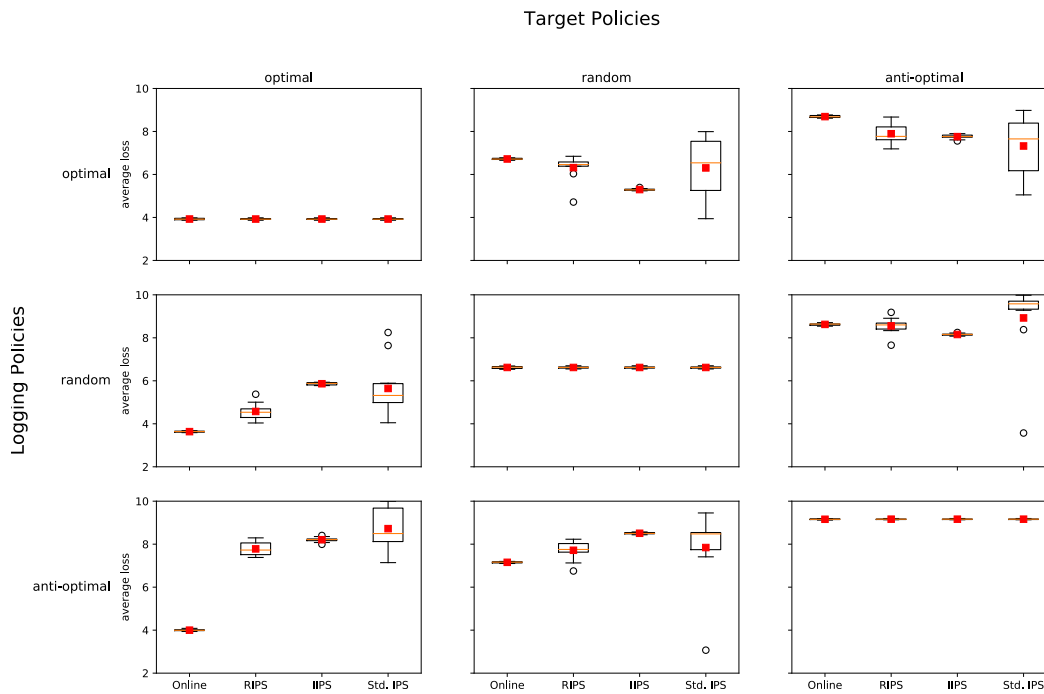
Target Policies

Figure 2: *Top*: **Results of the simulation experiment showing the average reward estimates for three different target policies computed using various off-policy estimators when using optimal policy as the logging policy.** *Middle:* **same as the top but with uniform random as the logging policy.** *Bottom:* **same as the top but with anti-optimal policy as the logging policy.**

4.1.2 User Simulation: We simulate a user examining a slate sequentially starting from the top then examining each sub-action at a time. The rewards on the sub-actions are assumed to be binary, i.e., the user provides a positive or negative reward for each sub-action. Further, the simulation assumes reward interactions, i.e., the reward on a sub-action is conditional on previous rewards and sub-actions on a given slate. In this work, we used a reward model that has a cascading effect. When a user provides a negative reward to a sub-action, it has a cascading effect resulting in a negative reward for the next sub-action in the slate. This user model is similar to the one used by Chandar and Carterette [3]. We choose a more extreme version of cascade behavior to illustrate the bias introduced by such a user behavior in existing estimators.

Note that our assumptions about the user model will affect the estimates provided by the counterfactual estimators. Therefore, we rely on real-world experiments (see Section 5) to validate these assumptions about the user model and in turn, validate the comparison of estimators.

4.1.3 Policy Simulation: We simulate three simple policies in our experiments: *optimal*, *anti-optimal*, and uniform random policy. Since the true reward for all the sub-actions for a context is known ahead of time, we simulate the *optimal* and *anti-optimal* policy by ordering the sub-actions based on true rewards as generated from Section 4.1.1. In other words, we sort the sub-actions (i.e., tracks) by the true reward (i..e, probability of stream) to get the optimal policy and reverse sort to get the anti-optimal policy. We

also experimented with different policies with varying degrees of true effectiveness and observed similar results.

4.2 Off-Policy Estimators

In our experiments, we compare our proposed off-policy estimator to various state-of-the-art method in the literature that are relevant to our problem. We briefly describe the estimators compared below:

- **Standard IPS**: takes the whole slate as an action and estimates the average reward. In our experiments, we use normalized importance sampling in which the weights are rescaled to sum to 1 in order to reduce variance [24].
- **Independent IPS (IIPS)**: Proposed by Li et al. [14], the estimator treats the reward at each position independently. The average reward of the target policy is estimated using Eq. 2.
- **Reward interaction IPS (RIPS)**: estimates the nested expectation by accumulating weights from the top of the slate to the bottom and estimates the average reward using Eq. 4.

4.3 How does reward interaction in user behavior affect off-policy estimation?

We evaluated the effectiveness of the three simulated policies—optimal, anti-optimal, and uniform random—using different off-policy estimators described in Section 4.2. Figure 2 shows the estimated average reward for the three target policies while using the logs from the different logging policies. The three rows in the figure represent the same experiment repeated with different policies as
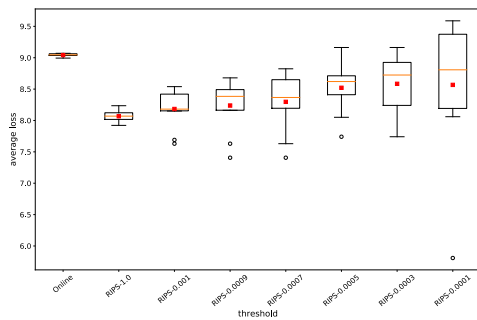
Figure 3: Performance of the RIPS estimator for different threshold values (closer to `Online` is better).

the logging policy, the diagonal represents the empirical average reward since the logging and target policy are the same. We expect the off-policy estimators to recover the true average reward of the policy as indicated by *online* scores in each of the plots. Each experiment was repeated 20 times and the error bars represent 95% confidence intervals.

We observed that the *Standard IPS* estimator has a large variance, making it less favorable to be used in practice. *IIPS*, on the other hand, has the least variance but provides a biased estimate whereas RIPS has considerably lower variance than *Standard IPS* and provides the most accurate estimate amongst the three.

4.4 How do *ESS* threshold and slate size affect the proposed RIPS estimator?

The threshold parameter t used in the RIPS estimator controls the lookback size for a given slate to mitigate extreme weighting. The threshold is a way of reducing the variance of the estimator while introducing a small amount of bias. To better understand the effect of the threshold parameter on the performance of RIPS estimator, we set up an experiment to evaluate RIPS under different threshold values. Figure 3 shows the average loss of the *anti-optimal* policy estimated by the RIPS estimator with different threshold values and compares it to the empirical average as indicated by *online*.

When threshold $t = 1.0$, RIPS decomposes to *IIPS* since there is no lookback, i.e., *line 12* in Algorithm 1 will never be true. Next, we increasingly vary the threshold [5] to demonstrate the effect on the mean and variance of the estimates. It can be observed that as we relax the threshold the RIPS estimator looks back further to consider the reward interaction. The reason for the increased variance is due to fewer data available which results in extreme weighting. Since the threshold parameter directly depends on the number of effective samples, we wish to explore methods to enable automatic tuning of the threshold parameter in the future.

Finally, we repeat the simulation experiment for different slate sizes with the random policy as the logging and optimal policy as the target policy. Figure 4 shows the average reward according to different estimators. When the slate size is one, all estimators behave similarly as there are no reward interactions in this case

---

[5]We ignore values $t = 1.0$ to $t = 0.001$ because at those thresholds effective sample size was zero resulting in the same average reward value as RIPS-1.0.

but as the slate size increases, we notice a gradual increase in the variance of the *Standard IPS estimator* and the bias in *IIPS* increases. RIPS is least affected by the slate size.

5 EXPERIMENTS ON REAL-WORLD DATA

In this section, we investigate how well various off-policy estimators perform at evaluating track sequencing algorithms used in an online environment to shuffle tracks within playlists on a large music streaming platform. An A/B test was run over a 6-week period in the summer of 2019 using 4 different shuffling algorithms: (1) a uniformly random shuffle, (2) a shuffle algorithm biased towards selecting more popular tracks first and which is non-personalized, (3) a personalized shuffle algorithm biased towards selecting tracks that the user has the highest affinity towards as predicted by a proprietary user-track affinity model, and (4) a shuffle algorithm biased towards selecting tracks acoustically similar to the previous track the user has heard. Note that policies (1), (2) & (3) do not depend on previous actions but (4) does. Approximately four million users, chosen at random, were exposed to one of these variants at random when commencing a session listening to one of a subset of 15 popular playlists available to all users on the platform. We define a session to be a continuous period of listening on a single playlist with no pause of greater than 60 seconds between playback of subsequent tracks.

In our study, we consider two primary metrics to compare different sequencing algorithms: skip rate and the listening time per session. Skip rate is the ratio of number of track skipped in a session to the number of tracks played and listening time is the total time spent in a session. Table 1 summarizes the online performances of the four policies tested in this study as characterized by the inverse of skip rate and listening time per session. As shown in the table, the affinity-biased approach has both the fewest skipped tracks in the first 10 of the session and the longest session length, followed closely by popularity-biasing. Acoustic biasing produces fewer skips than uniform random shuffling, but also slightly shorter listening sessions.

5.1 Comparison of Off-Policy Estimators

We measure how well each estimator predicts the online outcomes, based on logged data collected under each of the four shuffling algorithms, i.e., *logging policies*. We use skip-rate as the primary metric to be predicted, cap the session length to a maximum of 10 tracks, and compute our estimates using data collected under the uniformly random logging policy. We evaluate our estimators in terms of RMSE with respect to the ground truth obtained on-policy during the A/B test, that is, RMSE computed over the four estimates produced by each estimator. We compare our estimator, RIPS, against baselines described in 4.2

Each of the off-policy estimators is compared using data collected under the uniformly random logging policy. Since deploying a uniform random logging policy for an extended period is highly undesirable due to the negative impact it has on business metrics, it is important to compare the effectiveness of the estimators under non-uniform logging policies too. Table 2 shows the RMSE for each estimator, we observe that RIPS outperforms the baselines and the advantage for RIPS is even greater under the non-uniform
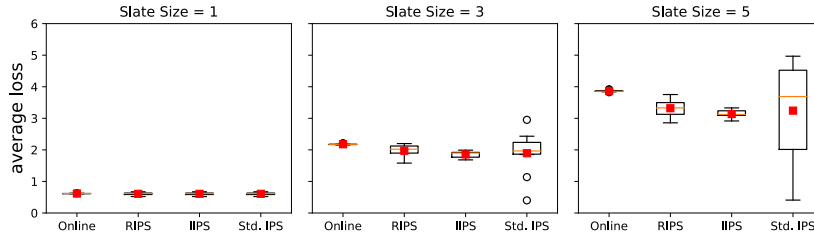
Figure 4: Performance of the estimators for different slate size.

Sequencer	Unskipped tracks in top 10	Session length	Number of users
Uniform random	5.86	701.7s	1, 012, 658
Popularity biased	6.00	726.9s	1, 016, 195
User affinity biased	6.09	733.3s	1, 010, 027
Acoustic biased	5.90	697.1s	998, 708

Table 1: Results and number of users for each sequencer in the online A/B test. Our primary metric is the number of unskipped tracks in the first 10 tracks of a session. All means for results significantly different at 0.05 level according to Tukey's HSD test for comparing pairs of means with corrections for multiple comparisons.

Logging policy	IPS	IIPS	RIPS (%Δ from IIPS)
Uniformly random	1.893	0.263	**0.194** (26%)
Popularity biased	1.812	0.681	**0.391** (43%)
User affinity biased	1.932	0.284	**0.196** (30%)
Acoustic biased	3.480	0.370	**0.364** (2%)

Table 2: RMSE of estimators under the different logging policies.

logging policies. The fact that IIPS shows a stronger decline in performance for the other logging policies may reflect that the sequential dependencies between rewards are stronger when the logging policy is not simply uniformly sampling from the available track pool but is instead biased towards a particular subset of the pool.

While Table 2 used the skip-rate metric to compare policies, we additionally compared the estimators using the listening time per session (in seconds) metric. The results followed a similar trend, we observed that the performance of RIPS was even more pronounced with an RMSE of **43.3** compared to IIPS (63.2) and IPS (303.0).

5.2 Impact of Dataset Size

We begin by investigating how the estimators perform for different dataset sizes. We vary the proportion of our dataset available to the estimators, from 1% of the data, corresponding to approximately 50,000 sessions, to 100% of the data, i.e. all 5 million sessions collected under the uniformly random logging policy. The results, shown in Figure 5, demonstrates the poor performance of IPS and that once we use more than 15% of the logged data, RIPS significantly outperforms IIPS. IIPS does not improve noticeably as the size of the dataset increases, suggesting that bias, rather than variance is the issue. Conversely, for IPS and RIPS, increasing the amount of data leads to better estimates. For the full dataset, RIPS obtains an RMSE of 0.194, compared to an RMSE of 0.263 for IIPS, and 1.893 for IPS. Note that the y axis is logarithmic to allow the results for IPS to be shown alongside those of IIPS and RIPS.
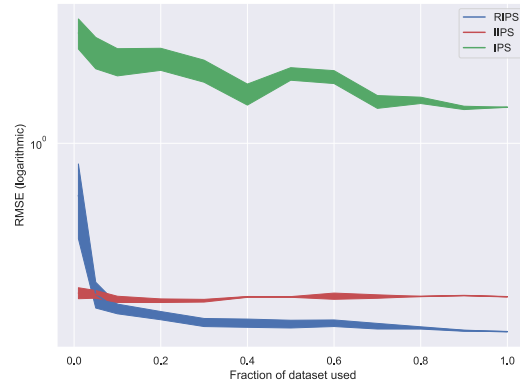

Figure 5: Performance of the estimators for different dataset sizes. Note that the y axis is logarithmic to allow the results for IPS to be shown clearly alongside the differences between IIPS and RIPS. The shaded areas indicate the maximum and minimum across 5 runs for each estimator

5.3 Impact of Slate Size

Since the propensities grow exponentially as the lengths of the slate size increase, an estimator like IPS is expected to perform poorly for longer sequences. When does this effect become prohibitive, and when do potential sequential dependencies have enough impact for IIPS to become biased? Figure 6 shows how the estimators perform when we consider different slate sizes (i.e., when we consider only those sessions with shorter than a given threshold number of tracks). We note that even for slate size as short as length 5, the RMSE for IPS begins to diverge significantly. For IIPS, the bias caused by the independence assumption does not result in substantially worse performance than for RIPS until we get to size 8 or longer.
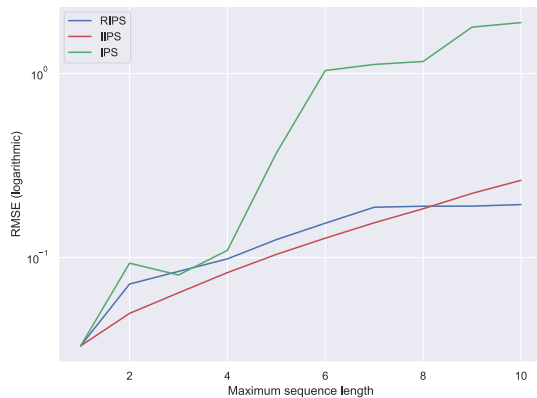
Figure 6: Performance of the estimators for different slate sizes (i.e., sequence lengths). Note again the log scale on the y axis.

5.4 Comparison with Pseudoinverse Estimator

For completeness, even though the assumptions of the pseudoinverse (PI) estimator are violated in our problem setting, as noted in Section 3.2.2, we include an experimental comparison with PI [25]. We use the simplified version described in Example 5 in [25], i.e., the case when the logging policy is uniform random, and the slate size is equal to the number of candidates. Here, we observe an RMSE of 0.738 for IIPS, 0.740 for PI, and **0.291** for RIPS. We omit the generalized form of PI proposed in [25] (Eq. 5 and 6), since there is no closed-form solution for computing Γ for our logging policies.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we presented RIPS, a new counterfactual estimator for sequential recommendations derived from a causal graph specification of the interactions between actions and rewards in a slate. Our approach is a lightweight non-parametric method that avoids the overhead of full parametric estimation associated with the direct method and is asymptotically unbiased.

Experiments using simulations and on a live system validate that RIPS has lower variance than IPS and lower bias than independent IPS when sequential reward interactions exists. This translates to a lower error when estimating slate-level skip rates and listening time for playlists. This improvement is maintained under different non-uniform logging policies, a key application for offline evaluation. Finally, our findings also indicate that significant sequential reward interaction exists among songs within playlists.

In future work, the methodology presented here may be extended to other kinds of reward interactions, e.g., a 2-dimensional grid of actions or longer-range dependencies. Another promising area to explore in the future merging our model agnostic approach with recent advances in doubly robust estimation such as [2, 23, 27]

ACKNOWLEDGMENTS

REFERENCES

[1] J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *Proceedings of the ICML*, pages 22–31, 2017.
[2] A. F. Bibaut, I. Malenica, N. Vlassis, and M. J. van der Laan. More efficient off-policy evaluation through regularized targeted learning. In *Proceedings of the ICML*, pages 654–663, 2019.
[3] P. Chandar and B. Carterette. Estimating clickthrough bias in the cascade model. In *Proceedings of CIKM*, CIKM '18, pages 1587–1590, 2018.
[4] M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi. Top-K off-policy correction for a REINFORCE recommender system. In *Proceedings of WSDM*, pages 456–464, 2019.
[5] M. Dudík, J. Langford, and L. Li. Doubly robust policy evaluation and learning. In *Proceedings of the ICML*, pages 1097–1104, 2011.
[6] A. Gilotte, C. Calauzènes, T. Nedelec, A. Abraham, and S. Dollé. Offline A/B testing for recommender systems. In *Proceedings of WSDM*, pages 198–206, 2018.
[7] C. A. Gomez-Uribe and N. Hunt. The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):13, 2016.
[8] L. Guo, H. Yin, Q. Wang, T. Chen, A. Zhou, and N. Quoc Viet Hung. Streaming session-based recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 1569–1577, New York, NY, USA, 2019. ACM.
[9] G. W. Imbens and D. B. Rubin. *Causal inference in statistics, social, and biomedical sciences.* Cambridge University Press, 2015.
[10] R. Jagerman, I. Markov, and M. de Rijke. When people change their mind: Off-policy evaluation in non-stationary recommendation environments. In *Proceedings of WSDM*, pages 447–455, 2019.
[11] L. Kish. *Survey sampling.* John Wiley and Sons, 1965.
[12] R. Kohavi, A. Deng, B. Frasca, T. Walker, Y. Xu, and N. Pohlmann. Online controlled experiments at large scale. In *Proceedings of ACM SIGKDD*, pages 1168–1176, 2013.
[13] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of WWW*, pages 661–670, 2010.
[14] S. Li, Y. Abbasi-Yadkori, B. Kveton, S. Muthukrishnan, V. Vinay, and Z. Wen. Offline evaluation of ranking policies with click models. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1685–1694, 2018.
[15] D. Liang, L. Charlin, J. McInerney, and D. M. Blei. Modeling user exposure in recommendation. In *Proceedings of WWW*, pages 951–961, 2016.
[16] M. Ludewig and D. Jannach. Evaluation of session-based recommendation algorithms. *CoRR*, abs/1803.09587, 2018.
[17] B. M. Marlin and R. S. Zemel. Collaborative prediction and ranking with non-random missing data. In *Proceedings of RecSys*, RecSys '09, pages 5–12, 2009.
[18] T. Nedelec, N. L. Roux, and V. Perchet. A comparative study of counterfactual estimators. *arXiv preprint arXiv:1704.00773*, 2017.
[19] J. Pearl. *Causality.* Cambridge university press, 2009.
[20] M. Sanderson. Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval*, 4(4):247–375, 2010.
[21] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the ICML*, pages 1670–1679, 2016.
[22] H. Steck. Evaluation of recommendations: Rating-prediction and ranking. In *Proceedings of RecSys'13*, pages 213–220, 2013.
[23] Y. Su, L. Wang, M. Santacatterina, and T. Joachims. CAB: Continuous adaptive blending for policy evaluation and learning. In *International Conference on Machine Learning*, pages 6005–6014, 2019.
[24] A. Swaminathan and T. Joachims. The self-normalized estimator for counterfactual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3231–3239, 2015.
[25] A. Swaminathan, A. Krishnamurthy, A. Agarwal, M. Dudik, J. Langford, D. Jose, and I. Zitouni. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3632–3642, 2017.
[26] J. Tang and K. Wang. Personalized top-N sequential recommendation via convolutional sequence embedding. In *Proceedings of WSDM*, WSDM '18, pages 565–573, 2018.
[27] P. Thomas and E. Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *Proceedings of the ICML*, pages 2139–2148, 2016.
[28] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He. A simple convolutional generative network for next item recommendation. In *Proceedings of WSDM*, WSDM '19, pages 582–590, 2019.

A PROPERTIES OF RIPS

The appendix provides a discussion of the properties of reward interaction inverse propensity scoring (RIPS). The pseudocode for

---

Algorithm 1: Reward interaction inverse propensity scoring (RIPS)

---

input : target policy h, logging policy π, threshold factor t, observed contexts $X^{(1:N)}$, actions $A_{1:K}^{(1:N)}$, and rewards $R_{1:K}^{(1:N)}$

1 **for** position $k \leftarrow 1$ **to** K **do**
2   **for** sample $n \leftarrow 1$ **to** N **do**
3   calculate independent weight $w_k^{(n)} \leftarrow \dfrac{h(A_k^{(n)} \mid X^{(n)})}{\pi(A_k^{(n)} \mid X^{(n)})}$
4   **end**
5   initialize lookback $b_k \leftarrow 0$
6   initialize RIPS weights $\gamma_k^{(1:N)} \leftarrow 1$
7   **while** $b_k < k$ **do**
8   **for** sample $n \leftarrow 1$ **to** N **do**
9   calculate proposal weight
         $u_k^{(n)} \leftarrow \dfrac{N \gamma_k^{(n)} w_{k-b_k}^{(n)}}{\sum_{n'=1}^{N} \gamma_k^{(n')} w_{k-b_k}^{(n')}}$
10   **end**
11   using Eq. 8:
12   **if**
       $\mathrm{ESS}(u_k^{(1:N)}) > Nt$ and $\mathrm{ESS}(u_k^{(1:N)}) < \mathrm{ESS}(\gamma_k^{(1:N)})$
       then $\gamma_k^{(1:N)} \leftarrow u_k^{(1:N)}$
13   **else** break
14      $b_k \leftarrow b_k + 1$
15   **end**
16 **end**
17 **return** target policy value estimate
     $\hat{V}(h) \leftarrow \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_k^{(n)} R_k^{(n)}$

RIPS is provided in Algorithm 1. First, the bias tends to zero as $N \to \infty$ assuming that the rewards have the conditional dependencies in Fig. 1c. Second, RIPS has lower variance than standard IPS with slate data in Eq. 1.

RIPS Unbiased in the Limit of Infinite Data: Need to prove that $\mathbb{E}[\gamma_{k-1}^{(n)} w_k^{(n)}] = 1$. Proof by induction:

- *Base case*, $k = 1$: $\mathbb{E}[1 \times w_k^{(n)}] = 1$ by the derivation in Eq. 5.
- *Inductive case*, $k \to k + 1$: recall that $\gamma_{k-1}^{(n)}$ is a random variable depending on $A_{1:k-1}$ and $w_k^{(n)}$ is a random variable depending on A_k, so make these dependencies explicit (for slate n),

$$\mathbb{E}[\gamma_{k-1}^{(n)}(A_{1:k-1}) w_k^{(n)}(A_k)] \tag{9}$$

$$= \int \gamma_{k-1}^{(n)}(A_{1:k-1}) w_k^{(n)}(A_k) \prod_{j=1}^{k} \pi(A_j \mid A_{j-1}, X) \mathrm{d}A_{1:k} \tag{10}$$

$$= \int \gamma_{k-1}^{(n)}(A_{1:k-1}) h(A_k \mid A_{k-1}, X) \prod_{j=1}^{k-1} \pi(A_j \mid A_{j-1}, X) \mathrm{d}A_{1:k} \tag{11}$$

$$= \int \gamma_{k-1}^{(n)}(A_{1:k-1}) \prod_{j=1}^{k-1} \pi(A_j \mid A_{j-1}, X) \left(\int h(A_k \mid A_{k-1}, X) \mathrm{d}A_k \right) \mathrm{d}A_{1:k-1} \tag{12}$$

$$= \int \gamma_{k-1}^{(n)}(A_{1:k-1}) \prod_{j=1}^{k-1} \pi(A_j \mid A_{j-1}, X) \mathrm{d}A_{1:k-1} \tag{13}$$

$$= 1. \tag{14}$$

where inductive hypothesis was used in the final line. ∎

By the strong law of large numbers, the sampled weights will be equal to the population mean as $n \to \infty$. In the limit, the weighting factor in Eq. 7 is equal to $\frac{1}{N}$ and $\hat{V}_{\mathrm{RIPS}}$ reduces to IPS which is unbiased.

Variance Properties of RIPS. We compare the variance of RIPS with that of the NIS applied to slate estimation [6]. NIS has favourable variance reduction properties and is the most suitable comparison due to its widespread use.

$$\hat{V}_{\mathrm{NIS}}(h) := \frac{1}{N} \sum_{n=1}^{N} \frac{\prod_{k=1}^{K} w_k^{(n)}}{\sum_{m=1}^{N} \prod_{k=1}^{K} w_k^{(m)}} \sum_{k=1}^{K} R_k^{(n)} \tag{15}$$

To aid the comparison, here is the closed-form RIPS expression,

$$\hat{V}_{\mathrm{RIPS}} = \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{\prod_{k'=1}^{k} w_{k'}^{(n)}}{\sum_{m=1}^{N} \prod_{k'=1}^{k} w_{k'}^{(m)}} R_k^{(n)}, \tag{16}$$

which can be seen by the fact that the denominator in Eq. 7 for position k cancels out the denominator for position $k - 1$ leaving only the sum-product of all the weights up to and including position k as the normalizer.

Here is a proof by induction that RIPS has non-strictly lower variance than NIS. Comparing Eq. 1 with Eq. 16, notice that both sum over the reweighted position-based rewards $R_k^{(n)}$ for $k = 1, \ldots, K$. We start at the last position and work backwards.

- *Base case*: starting at position K in the slate, the reweighting factor for both RIPS and NIS is $\prod_{k=1}^{K} w_k^{(n)}$, therefore the variance of the reward estimator for the last position is identical.
- *Inductive case*: moving from position $k \to k - 1$, observe that the number of reweighting factors in RIPS reduces by 1 while the number of reweighting factors in NIS remains constant. By the fact that $\mathrm{Var}\left[\frac{h(A_k \mid X)}{\pi(A_k \mid X)} \right] \geq 0$, the variance of RIPS must be strictly less than that of NIS.

Further, RIPS has a strictly lower variance than NIS if any of the reweighting factors $k < K$ has non-zero variance, which is commonly the case. ∎

Finally, as $N \to \infty$, the effective sample size also goes to infinity if absolute continuity between the logging and target policies holds.