Variance-Minimizing Augmentation Logging for Counterfactual Evaluation in Contextual Bandits

Aaron D. Tucker
aarondtucker@cs.cornell.edu
Cornell University
Ithaca, New York, USA

Thorsten Joachims
tj@cs.cornell.edu
Cornell University
Ithaca, New York, USA

ABSTRACT

Methods for offline A/B testing and counterfactual learning are seeing rapid adoption in search and recommender systems, since they allow efficient reuse of existing log data. However, there are fundamental limits to using existing log data alone, since the counterfactual estimators that are commonly used in these methods can have large bias and large variance when the logging policy is very different from the target policy being evaluated. To overcome this limitation, we explore the question of how to design data-gathering policies that most effectively augment an existing dataset of bandit feedback with additional observations for both learning and evaluation. To this effect, this paper introduces Minimum Variance Augmentation Logging (MVAL), a method for constructing logging policies that minimize the variance of the downstream evaluation or learning problem. We explore multiple approaches to computing MVAL policies efficiently, and find that they can be substantially more effective in decreasing the variance of an estimator than naïve approaches.

CCS CONCEPTS

• **Computing methodologies** → **Batch learning**; **Active learning settings**; • **Theory of computation** → **Sequential decision making**.

KEYWORDS

counterfactual inference, off-policy evaluation and learning, contextual bandits, reinforcement learning, recommender systems

ACM Reference Format:
Aaron D. Tucker and Thorsten Joachims. 2023. Variance-Minimizing Augmentation Logging for Counterfactual Evaluation in Contextual Bandits. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27-March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570452

1 INTRODUCTION

Logged user feedback from online systems is one of the primary sources of training data for search and recommender systems. However, learning from log data is challenging since the rewards are only partially observed. In particular, logged data contains the observed reward (e.g. click/no click) only for the specific action (e.g. movie recommendation) that the historic system took, but logs do not include the reward observations for the other possible actions the system could have taken (e.g. all other movies). This means that offline policy evaluation requires us to deal with counterfactual outcomes when the historic system and the new policy do not chose the same action.

Fortunately, over the recent years an increasingly rich set of counterfactual estimators [6, 10, 11, 22, 24, 26, 28, 30] and learning methods [4, 14, 19, 23–25] have been developed that can use logged user feedback with strong theoretical guarantees despite the partial nature of the data. These developments have led to increasing adoption of counterfactual learning and evaluation in real-world applications, where they are used to conduct "offline A/B tests" and to train policies that directly and provably optimize online metrics. However, we point out that all counterfactual methods are fundamentally limited by the information contained in the logs. In particular, if the policy that logged the data is substantially different from the target policy we want to evaluate, the variance of the estimate will be high or the estimator may even be biased [e.g., 22].

To overcome this fundamental limitation of counterfactual methods, this paper explores how to best collect a limited amount of additional log data to maximize the effectiveness of the counterfactual estimator. We call this the problem of *augmentation logging*, and we study how to design augmentation logging policies that optimally augment an existing log dataset. The resulting methods can be used to optimize data efficiency for A/B testing, and even address the question of how to log data in contextual-bandit systems that are re-trained periodically (e.g., weekly).

The main contributions of this paper are four-fold. First, we introduce and formalize the problem of augmentation logging as minimizing the bias and variance of the counterfactual estimator given existing logged data. Second, we show how to compute variance-optimal augmentation policies and provide a theoretical characterization of this approach. Third, we develop a method to approximate the optimal augmentation policy, improving its online efficiency. Finally, we empirically evaluate the methods for both counterfactual policy evaluation and counterfactual policy learning.

2 RELATED WORK

While our formulation of optimal augmentation logging for counterfactual policy evaluation and learning is novel, it is connected to several bodies of existing literature.

Off-policy evaluation. Counterfactual evaluation (a.k.a. off-policy evaluation, offline evaluation, or offline A/B testing) has been widely

WSDM '23, February 27-March 3, 2023, Singapore, Singapore

ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570452

studied as a method of estimating the value of a policy based on data collected under a different policy. Estimators like Inverse Propensity Score (IPS) weighting and its variants [13, 23, 26] are typically used to correct the distribution shift between logging and target policy. While unbiased under full support [21, 23], such estimators can have large variance. Our method aims to reduce the variance of the estimator, which is also the motivation behind work such as Bottou et al. [6], Dudík et al. [10], Farajtabar et al. [11], Sachdeva et al. [22], Su et al. [24], Wang et al. [30], or Thomas and Brunskill [28] for evaluation; and behind Bottou et al. [6], Joachims et al. [14], Strehl et al. [23], Su et al. [24], Swaminathan and Joachims [25], or London and Sandler [19] for learning. However, instead of treating the data as fixed and trying to reduce bias and variance with this constraint, we investigate which additional data would most reduce bias and variance.

Batched Bandits. Online contextual bandits (see e.g. Cesa-Bianchi and Lugosi [7]) have been studied as a model of sequential decision making under a variety of settings and modeling assumptions [2, 3, 5, 16, 18]. However, as other researchers have noted, the ability to change the policy for each context is not necessarily realistic, motivating the creation of Batched Bandits [12]. Our method differs from work on Batched Bandits in that our method minimizes the maximum variance of the off-policy estimate for any $\pi \in \Pi$, for any chosen policy class Π. Being able to choose the policy class Π is valuable since users can impose whatever constraints they want on Π, such as fairness. Secondly, our method allows for substantially more flexible policy classes than are typical in the batched bandits literature, and we show that it can effectively learn policies that are parameterized by neural networks. Thirdly, our method allows for arbitrary policy histories. Instead of assuming that the bandit algorithm controls the full sequence of policies, our method works after any sequence of known policies.

Active Learning. Augmentation logging also shares similarities with Active Learning methods, which seek to prioritize the collection of the most informative data. Of particular note is a line of research starting at CORNUET et al. [9], which combines multiple importance samplers with an adaptive setting. More recent work in this area including Yan et al. [31] and Yan et al. [32] are quite similar to our method in that they also use a weighted combination of two different importance weights in order to reduce the variance of their estimates. However, they differ in a few crucial regards. The biggest difference is that these methods focus on full-information classification rather than partial-information policy evaluation and learning. Where their goal is to find a classifier that has low error over $x \sim \mathcal{D}$, our goal is to find a policy which maximizes the expected reward in bandit settings. As part of this, our method is able to handle reward values in $\mathbb{R}$, rather than a binary or categorical label space $\mathcal{Y}$.

Monte-Carlo Estimation. Designing optimal sampling distributions is a problem widely considered in Monte-Carlo estimation [21]. We draw upon foundational results about which sampling strategies are optimal for importance sampling estimators, which are analogous to IPS estimators. Moreover, we relate augmentation logging to multiple importance sampling [1], and we show how to extend these methods to get uniform bounds on the variance of a class of target policies.

3 AUGMENTATION LOGGING FOR SINGLE-POLICY EVALUATION

We begin by formalizing the augmentation-logging problem for evaluating a single target policy π_{tar}, and then extend this approach to multi-policy evaluation and learning in Section 4. In all three settings, we consider contextual bandit policies, which are widely used to model search and recommendation problems [17, 18]. At each time step i, a context x_i (e.g., query, user request) is sampled i.i.d. from an underlying distribution $x_i \sim \Pr(X)$, and a policy π stochastically chooses an action a_i (e.g., a movie to recommend) such that $a_i \sim \pi(A|x)$. The system then observes the reward r_i (e.g., purchase) for action a_i from the environment.

The central question in single-policy evaluation lies in estimating the expected reward (a.k.a. utility)

$$R_{\pi_{\text{tar}}} = \sum_x \sum_a \mathbb{E}_r[r(x,a)] \pi_{\text{tar}}(a|x) \Pr(x) \tag{1}$$

of some target policy π_{tar}. The conventional approach is to field this target policy in an A/B test, which allows us to estimate $U(\pi_{\text{tar}})$ simply from the average of the observed rewards. However, such online A/B tests typically take a long time to complete, and they do not scale when we need to evaluate many target policies. Therefore, offline evaluation has seen substantial interest, since it computes an estimate of $U(\pi_{\text{tar}})$ using only historic data

$$\mathcal{D}_{\text{log}} = \{x_i, a_i, r_i\}_{i=1}^{n_{\text{log}}} \tag{2}$$

already logged from some other policy π_{log}.[1] The key challenge lies in the fact that the logging policy π_{log} typically picks actions that are different from those selected by the target policy π_{tar}. This challenge can be addressed by counterfactual estimators such as inverse propensity score (IPS) weighting

$$R^{\text{IPS}}_{\pi_{\text{tar}}} = \frac{1}{n_{\text{log}}} \sum_{i=1}^{n_{\text{log}}} \frac{\pi_{\text{tar}}(a_i|x_i)}{\pi_{\text{log}}(a_i|x_i)} r_i. \tag{3}$$

The IPS estimator can be shown to be unbiased whenever the logging policy has full support under the target policy, namely when $\forall x \forall a : \pi_{\text{tar}}(a|x)P(x) > 0 \rightarrow \pi_{\text{log}}(a|x) > 0$. Unfortunately, this condition is frequently violated in practical applications. Further, even if the condition is met, the IPS estimator can have excessive variance. Much work has gone into mitigating both the bias problem [10, 22] and the variance problem [6, 10, 24, 28, 30], but any estimator that only has the information in $\mathcal{D}_{\text{log}}$ is fundamentally limited.

To overcome these fundamental limits, we allow that we can augment $\mathcal{D}_{\text{log}}$ with n_{aug} additional observations

$$\mathcal{D}_{\text{aug}} = \{x_i, a_i, r_i\}_{i=1}^{n_{\text{aug}}} \tag{4}$$

from an augmentation logging policy π_{aug}. We next address the key questions of which counterfactual estimator to use, and how to design π_{aug} so that the n_{aug} additional observations most improve the quality of the utility estimate.

[1] For simplicity of notation, we assume all observations were collected from the same π_{log}. However, all results in this paper can be extended to the case where the data comes from multiple logging policies.

3.1 Designing Variance-Optimal Augmentation Policies

In order to reason about which augmentation policy π_{aug} minimizes bias and variance, we first need to select an estimator. As we will justify below, we focus on the balanced estimator [1],

$$\hat{R}^{\text{BAL}}_{\pi_{\text{tar}}} = \frac{1}{N}\left(\sum_{(x_i,a_i,r_i)\in\mathcal{D}_{\text{log}}\cup\mathcal{D}_{\text{aug}}} \frac{\pi_{\text{tar}}(a_i|x_i)}{\pi_{\text{balanced}}(a_i|x_i)} r_i\right), \tag{5}$$

where $N = n_{\text{log}} + n_{\text{aug}}$, $\alpha = n_{\text{aug}}/N$ and

$$\pi_{\text{balanced}}(a|x) \quad = \quad (1-\alpha)\pi_{\text{log}}(a|x) + \alpha\pi_{\text{aug}}(a|x).$$

This estimator was shown to never have a larger variance than the following more naïve IPS estimator

$$\hat{R}^{\text{IPS}}_{\pi_{\text{tar}}} = \frac{1}{N}\sum_{(x,a,r)_i\in\mathcal{D}_{\text{log}}} \frac{\pi_{\text{tar}}(a_i|x_i)}{\pi_{\text{log}}(a_i|x_i)} r_i + \frac{1}{N}\sum_{(x,a,r)_j\in\mathcal{D}_{\text{aug}}} \frac{\pi_{\text{tar}}(a_j|x_j)}{\pi_{\text{aug}}(a_j|x_j)} r_j$$

that weights each action by the policy that selected it, and it can have substantially smaller variance [1]. Further, it is easy to verify that the balanced estimator is unbiased under strictly weaker conditions than the IPS estimator. In particular, the balanced estimator is already unbiased if $\forall x \forall a : \pi_{\text{tar}}(a|x)P(x) > 0 \rightarrow (\pi_{\text{log}}(a|x) > 0 \vee \pi_{\text{aug}}(a|x) > 0)$. The balanced estimator has the following variance as shown in [1], with proof in Appendix A.1 as well.

$$\text{Var}\left[\hat{R}^{\text{BAL}}_{\pi_{\text{tar}}}\right] = \frac{1}{N}\left(\mathbb{E}_x\left[\sum_{a\in\mathcal{A}} \frac{\pi^2_{\text{tar}}(a|x)\mathbb{E}_r\left[r^2(x,a)\right]}{\pi_{\text{balanced}}(a|x)}\right] - R^2_{\pi_{\text{tar}}}\right) \tag{6}$$

In this equation, $\mathbb{E}_r\left[r^2(x,a)\right] = \bar{r}^2(x,a) + \sigma^2(x,a)$, where $\bar{r}(x,a)$ and $\sigma^2(x,a)$ are the expected rewards and their variance conditioned on the given x, a.

Importantly, this variance depends directly on π_{balanced}, which is based on π_{log} and π_{aug}. This allows us to design an augmentation policy π_{aug} which compensates for high variance terms caused by the logging policy π_{log}. This compensation is not possible for the naïve IPS estimator $\hat{R}^{\text{IPS}}_{\pi_{\text{tar}}}$, since the partial derivative of the variance with respect to $\pi_{\text{aug}}(a|x)$ does not include any terms with $\pi_{\text{log}}(a|x)$, nor are there any interactions in the constraints $\forall x : \sum_{a\in\mathcal{A}} \pi_{\text{aug}}(a|x) = 1$. Therefore the augmentation policy π_{aug} does not depend in any way on logging policy π_{log}, and the variance minimizing augmentation logging policy would be the same as if the logging policy π_{log} had not been used.

Given these limitations of the IPS estimator, we thus focus on the balanced estimator for designing augmentation logging policies. Note that it is straightforward to extend the balanced estimator to a doubly-robust setting [10], which we omit for the sake of brevity and to highlight the structural improvements provided by augmentation logging rather than other variance reduction techniques. We formulate the search for the variance minimizing augmentation policy as the following optimization problem, which we refer to as Minimum Variance Augmentation Logging (MVAL).

Optimization Problem 1 (MVAL for Single-Policy Evaluation). *For a given context $x \in \mathcal{X}$,*

$$\begin{aligned} \pi_{aug}(A|x) = \underset{\pi\in\mathcal{R}^{|\mathcal{A}|}}{\arg\min} \quad & \sum_{a\in\mathcal{A}} \frac{\pi^2_{tar}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]}{(1-\alpha)\pi_{log}(a|x) + \alpha\pi(a)} \\ \textit{subject to} \quad & \sum_{a\in\mathcal{A}} \pi(a) = 1, \\ & \forall a \in \mathcal{A} : \pi(a) \geq 0 \end{aligned}$$

The augmentation policy $\pi_{\text{aug}}(A|x)$ computed by OP1 minimizes the variance of the balanced estimator, since $\pi_{\text{balanced}}(a|x) = (1-\alpha)\pi_{\text{log}}(a|x) + \alpha\pi(a)$. Variance is a weighted sum of these independent minimized terms, and the number of real-valued parameters $\pi(a)$ in this optimization problem always equals the number of actions for the given context x.

The first key property of this optimization problem is that it is always convex, making it possible to efficiently find the solution.

Theorem 3.1 (MVAL Convexity). *OP1 is convex.*

Proof. This can be seen by taking the partial derivative of the variance of the balanced estimator in Equation (6) with respect to $\pi_{\text{aug}}(a|x)$ is

$$\frac{\partial}{\partial\pi_{\text{aug}}(a|x)}\text{Var}\left[\hat{R}^{\text{BAL}}_{\pi_{\text{tar}}}\right] = -\frac{\alpha}{N}\frac{\pi^2_{\text{tar}}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]\Pr(x)}{\Big((1-\alpha)\pi_{\text{log}}(a|x) + \alpha\pi_{\text{aug}}(a|x)\Big)^2}.$$

This partial derivative is always negative, because there is a negative sign, and $\pi(a|x) > 0, \pi^2(a|x) > 0$ for all valid π, and $\bar{r}^2(x,a)$ and $\sigma^2(x,a)$ are also always positive.

Now consider its second derivatives. For $x', a' \neq x, a$,

$$\frac{\partial^2}{\partial\pi_{\text{aug}}(a|x)\partial\pi_{\text{aug}}(a'|x')}\text{Var}\left[\hat{R}^{\text{BAL}}_{\pi_{\text{tar}}}\right] = 0.$$

Because these are all zero, the Hessian matrix of the variance of the balanced estimator is diagonal. For x, a,

$$\frac{\partial^2}{\partial^2\pi_{\text{aug}}(a|x)}\text{Var}\left[\hat{R}^{\text{BAL}}_{\pi_{\text{tar}}}\right] = \frac{2\alpha^2}{N}\frac{\pi^2_{\text{tar}}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]\Pr(x)}{\Big((1-\alpha)\pi_{\text{log}}(a|x) + \alpha\pi_{\text{aug}}(a|x)\Big)^3}.$$

Note that every term of this equation is always positive, so this term is always positive. It follows that every term in the diagonal of the Hessian is positive, and since the Hessian is a diagonal matrix, this means that the Hessian is positive-definite, and therefore the optimization problem is convex. □

The second key property of this optimization problem is that the augmentation policy π_{aug} it computes is guaranteed to produce data that makes the balanced estimator unbiased. Specifically, with augmentation data from π_{aug} the balanced estimator is unbiased, even if the logging policy π_{log} is support deficient and would otherwise lead to biased estimates.

Theorem 3.2 (MVAL Guarantees Unbiasedness). *OP1 always produces augmentation policies π_{aug} so that the balanced estimator is unbiased for any π_{log} and for any choice of $\bar{r}(x,a)$ and $\sigma^2(x,a) > 0$, even if $\bar{r}(x,a)$ and $\sigma^2(x,a)$ are inaccurate.*

Proof. As shown in [1], the balanced estimator is unbiased when π_{balanced} has full support for π_{tar}, specifically

$$\forall x \forall a : \pi_{\text{tar}}(a|x)P(x) > 0 \rightarrow \pi_{\text{balanced}}(a|x) > 0.$$

Since π_{balanced} is a convex combination of $\pi_{\log}$ and π_{aug}, full support is already guaranteed if $\forall x \forall a : \pi_{\text{tar}}(a|x)P(x) > 0 \rightarrow (\pi_{\log}(a|x) > 0 \vee \pi_{\text{aug}}(a|x) > 0)$. To show that this condition is always fulfilled, we need to verify that $\pi_{\text{aug}}(a|x) > 0$ when $\pi_{\log}(a|x) = 0$. To verify this condition, note that $\pi_{\log}(a|x) = 0$ implies that the term

$$\frac{\pi^2_{\text{tar}}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]}{\alpha\pi_{\text{aug}}(a)}$$

occurs in the objective of OP1. Note that the solution of OP1 cannot have $\pi_{\text{aug}}(a) = 0$, since this would lead to an infinite objective which is not optimal since the uniform π is feasible and has a better objective value. □

The final issue we need to resolve is that $\mathbb{E}_r\left[r^2(x,a)\right]$ is typically unknown. Fortunately, there are at least two options for handling this. The first option is to optimize the following variant of the MVAL optimization problem, where we simply drop $\mathbb{E}_r\left[r^2(x,a)\right]$ from the objective. This is equivalent to minimizing an upper bound on the variance, where $\mathbb{E}_r\left[r^2(x,a)\right]$ is replaced with $\max_{x,a}\mathbb{E}_r\left[r^2(x,a)\right]$ in Equation (6). This is the approach taken in our experiments unless otherwise noted. The second option is to use a regression estimate to impute the estimated value for $\mathbb{E}_r\left[r^2(x,a)\right]$. Virtually any real-valued regression technique can be applied to $\mathcal{D}_{\log}$ to estimate $r^2(x,a)$, and even imperfect estimates can provide useful information about $\text{Var}[\hat{R}^{\text{BAL}}_{\pi_{\text{tar}}}]$. Note that Theorem 3.2 holds even for incorrect estimates of $\mathbb{E}_r\left[r^2(x,a)\right]$ so long as the resulting augmentation policy $\pi_{\text{aug}}(a) > 0$, so bad estimates of $\mathbb{E}_r\left[r^2(x,a)\right] > 0$ only increase the variance of the estimates and never introduce bias.

3.2 Analysis and Discussion

We now further analyze the properties of MVAL policies and provide intuition through some illustrative edge cases.

3.2.1 MVAL without Historic Log Data. When the historic data $\mathcal{D}_{\log}$ is empty, the MVAL policy $\pi^{\text{BAL}}_{\text{aug}}(a|x)$ computed by OP1 coincides with the variance minimizing logging policy $\pi^{\text{IPS}}_{\text{minvar}}(a|x)$ for the IPS estimator. If $\alpha = n_{\text{aug}}/(n_{\text{aug}} + n_{\log}) = 1$, then the optimal augmentation policy is

$$\begin{aligned}\pi^{\text{BAL}}_{\text{aug}}(a|x) &= \frac{\pi_{\text{tar}}(a|x)\sqrt{\mathbb{E}_r\left[r^2(x,a)\right]}}{\sum\limits_{a\in\mathcal{A}}\pi_{\text{tar}}(a|x)\sqrt{\mathbb{E}_r\left[r^2(x,a)\right]}}\\ &= \pi^{\text{IPS}}_{\text{minvar}}(a|x).\end{aligned}$$

This can be shown using Lagrange multipliers to solve OP1, and using the well known result [21] characterizing the variance minimizing IPS logging policy. This immediately implies that if there is no logged data so that $\alpha = 1$, and no information about the mean and variance of the reward such that $\mathbb{E}_r\left[r^2(x,a)\right] = c$, then since the optimal augmentation policy $\pi^{\text{BAL}}_{\text{aug}}(a|x) \propto \pi_{\text{tar}}(a|x)\sqrt{\mathbb{E}_r\left[r^2(x,a)\right]}$, and $\mathbb{E}_r\left[r^2(x,a)\right] = c$, then $\pi^{\text{BAL}}_{\text{aug}}(a|x) \propto \pi_{\text{tar}}(a|x)$, and therefore the optimal augmentation policy for single policy evaluation using the balanced estimator coincides with the target policy.

3.2.2 MVAL Corrects Historic Log Data Towards the Target Policy. When the historic data is non-empty, the augmentation policy π_{aug} minimizes the balanced estimator variance by trying to make the balanced policy π_{balanced} similar to the minimum variance IPS policy $\pi^{\text{IPS}}_{\text{minvar}}$. Specifically, when there is enough augmentation data for π_{aug} to cause $\pi_{\text{balanced}} = \pi^{\text{IPS}}_{\text{minvar}}$, then that is the solution chosen.

In this case, we can even compute the MVAL policy in closed form. If α is big enough that the mixed policy $\pi_{\text{balanced}} = (1-\alpha)\pi_{\log} + \alpha\pi_{\text{aug}}$ can be equal to the minimum variance augmentation policy for the IPS estimator $\pi^{\text{IPS}}_{\text{minvar}}$, then the MVAL policy for the balanced estimator is the policy π such that $\forall x \in \mathcal{X}, a \in \mathcal{A} : (1-\alpha)\pi_{\log}(a|x) + \alpha\pi(a|x) = \pi^{\text{IPS}}_{\text{minvar}}(a|x)$. This is because the balanced estimator variance is the IPS variance with π_{balanced} instead of $\pi_{\log}$, so if π_{aug} can make $\pi_{\text{balanced}} = \pi^{\text{IPS}}_{\text{minvar}}$ then that is optimal.

The fact that, in the case of constant $\mathbb{E}_r\left[r^2(x,a)\right]$, MVAL aims to augment the existing data $\mathcal{D}_{\log}$ so that the overall data looks like it was all sampled from π_{tar} has an interesting implication for the overall utility during data collection. In particular, MVAL will ensure an overall utility as if all of $\mathcal{D}_{\log}$ and $\mathcal{D}_{\text{aug}}$ had been sampled from π_{tar}. Since it is the prior belief in many A/B tests that the target policy π_{tar} is better than the logging policy $\pi_{\log}$, this means that MVAL will improve utility during data collection in addition to sampling the most informative data.

3.2.3 Introducing a New Action. A common way a new target policy is different from the logging policy is through the introduction of a new action (e.g., a new movie). In this case, MVAL's behavior matches the intuition that this new action should now be sampled by the augmentation policy. This is because of Theorem 3.2, which states that MVAL produces unbiased estimates for any x such that $\mathbb{E}_r\left[r^2(x,a)\right] > 0$. If $\pi_{\text{tar}}(a|x) > 0 = \pi_{\log}(a|x)$ and $0 = \pi_{\text{aug}}(a|x)$, then it would be a biased estimate. This means that if there is an action $a \in \mathcal{A}$ such that $\pi_{\text{tar}}(a|x) > 0$ but $\pi_{\log}(a|x) = 0$, then the variance minimizing π_{aug} is such that $\pi_{\text{aug}}(a|x) > 0$.

3.2.4 Deterministic Target Policies. If π_{tar} is deterministic, then the optimal augmentation policy for single policy evaluation using the balanced estimator is π_{tar}. If $\pi_{\text{tar}}(a|x) = 0$ for some action, then that action contributes nothing to the balanced estimator variance, and since the variance contribution for each action decreases with more probability, using the deterministic target policy is optimal.

3.2.5 Variance Reduction through Augmentation Logging. One final point is that adding even a single augmentation data point can substantially decrease the estimator's variance. While the variance reduction of a single point depends on the specific logging and target policies, there is no upper bound on the variance reduction achievable by adding a single augmentation sample.

Consider that when the logging policy assigns almost no probability to an action with a large probability under the target policy, then that action has an arbitrarily large variance contribution. An action/context pair's contribution to the variance for a given logging policy is proportional to $\frac{1}{\pi_{\log}(a|x)}\left(\pi^2_{\text{tar}}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]\right)$. Since the last term is always positive, we can roughly note that changing from a logging policy where $\pi_{\text{log}}(a|x) = \epsilon$ to a balanced policy where $\pi_{\text{balanced}}(a|x) = \frac{N\epsilon+1}{N}$ results in roughly a $\left(\frac{1}{\epsilon} - \frac{N}{N\epsilon+1}\right)\left(\pi^2_{\text{tar}}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]\right)$ variance improvement. As $\epsilon \to 0$, this expression goes to ∞.

3.3 Pre-Computing MVAL Policies

So far we have assumed that we simply solve the MVAL optimization problem for each individual context x as it comes in. This is realistic in most applications, since these optimization problems are convex (see Theorem 3.1) and no bigger than the number of available actions. However, some applications may have latency requirements where even this modest amount of computation is not feasible. We therefore ask whether we can learn a general MVAL policy that applies to any context x ahead of time, so that this policy merely needs to be executed during deployment.

We approach the problem of learning a general MVAL policy as the following optimization problem. Given a parameterized space Π of candidate augmentation policies (e.g. deep network policies [14]) and a sample of contexts $\{x_i\}_{i=1}^N$, find the augmentation policy $\pi_{\text{aug}} \in \Pi$ that minimizes the sum of the variances over all N sample contexts.

Optimization Problem 2 (Pre-Computed MVAL Policy).

$$\begin{aligned}\underset{\pi\in\Pi}{\arg\min}\quad & \sum_{x_i\in\mathcal{D}}\sum_{a\in\mathcal{A}} \frac{\pi^2_{tar}(a|x_i)\,\mathbb{E}_r\left[r^2(x_i,a)\right]}{(1-\alpha)\pi_{log}(a|x_i)+\alpha\pi(a|x_i)}\\ \textit{subject to}\quad & \sum_{a\in\mathcal{A}} \pi(a|x_i) = 1,\\ & \pi(a|x_i) \geq 0 \textit{ for all } a \in \mathcal{A}, x_i \in \mathcal{D}\end{aligned}$$

OP2 constructs a policy by using empirical risk minimization on a sampled dataset with the MVAL objective. As is standard in ERM, an augmentation policy that minimizes the objective (here variance) on a large sample of training points can be expected to also produce good objective values on new contexts under standard conditions on the capacity of Π. We will empirically compare these pre-computed MVAL policies to on-the-fly computed MVAL policies in the experiments.

Note that while the optimal augmentation policy in this setup depends on the past policies, the target policy, the observed x_i, and the expected squared reward $\mathbb{E}_r\left[r^2(x_i,a)\right]$, it does not depend on any actions sampled under the past policies. This means that the contributed variance terms are independent, and OP2 does not violate any independence conditions.

4 AUGMENTATION LOGGING FOR MULTI-POLICY EVALUATION AND LEARNING

The previous section showed how to optimally augment an existing dataset $\mathcal{D}_{\text{log}}$ when evaluating a single target policy π_{tar}. However, in many practical offline A/B tests we may want to evaluate a set of competing target policies $\Pi_{\text{tar}} = \{\pi_1, \ldots \pi_k\}$, especially when we want to learn a new policy π^* through Empirical Risk Minimization (ERM) using the balanced estimator:

$$\pi^* = \underset{\pi\in\Pi_{\text{tar}}}{\arg\max}\,\hat{R}_\pi^{\text{BAL}} \tag{7}$$

We therefore ask the question of how to compute an MVAL policy that minimizes the maximum variance for any target policy in Π_{tar}

$$\pi_{\text{aug}}(A|x) = \underset{\pi_{\text{aug}}}{\arg\min}\,\max_{\pi\in\Pi_{\text{tar}}} \text{Var}\left[\hat{R}_\pi^{\text{BAL}}(x)\right], \tag{8}$$

where $R_\pi^{\text{BAL}}(x)$ is the estimate of the expected reward of the target policy in context x.

While solving Equation (8) directly can be challenging, the following upper bound on the variance for any π_{tar} in a class of policies Π_{tar} leads to an optimization problem that is no more complex than single policy evaluation beyond the calculation of $\max_{\pi\in\Pi_{\text{tar}}} \pi(a|x)$.

Theorem 4.1 (Policy Class Variance Bound). *Given a class of policies Π, then $\forall \pi \in \Pi$,*

$$Var\left[\hat{R}_\pi^{BAL}\right] \leq \frac{1}{N}\mathbb{E}_x\left[\sum_{a\in\mathcal{A}} \frac{\pi^2_{max}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]}{\pi_{balanced}(a|x)}\right].$$

where $\pi_{max}(a|x) = \max_{\pi\in\Pi}\pi(a|x)$.

Proof. By the definition, for all $\pi \in \Pi$,

$$\pi(a|x) \leq \max_{\pi'\in\Pi}\pi'(a|x) = \pi_{\max}(a|x).$$

For all a, x, since $\sigma^2(x,a)$, $r^2(x,a)$, and $\pi_{\text{balanced}}(a|x)$ are all positive,

$$0 \leq \frac{\mathbb{E}_r\left[r^2(x,a)\right]}{\pi_{\text{balanced}}(a|x)}.$$

Therefore, for all $\pi \in \Pi$, and all a, x,

$$\frac{\pi^2(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]}{\pi_{\text{balanced}}(a|x)} < \frac{\pi^2_{\text{max}}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]}{\pi_{\text{balanced}}(a|x)}.$$

Therefore, for all $\pi \in \Pi$,

$$\begin{aligned}\text{Var}\left[\hat{R}_\pi^{\text{BAL}}\right] &= \frac{1}{N}\mathbb{E}_x\left[\sum_{a\in\mathcal{A}} \frac{\pi^2(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]}{\pi_{\text{balanced}}(a|x)}\right]\\ &\leq \frac{1}{N}\mathbb{E}_x\left[\sum_{a\in\mathcal{A}} \frac{\pi^2_{\text{max}}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]}{\pi_{\text{balanced}}(a|x)}\right]\end{aligned}$$

□

The structure of this bound results in the same optimization problem as the one for single-policy optimization. In particular, the augmentation policy that minimizes the maximum value of the variance bound for any $\pi \in \Pi$ can be found by solving the following optimization problem, where $\pi_{\max}$ replaces the π_{tar} of OP1.

Optimization Problem 3 (MVAL for Multi-Policy Evaluation).

$$\begin{aligned}\underset{\pi_x\in\mathcal{R}^{|\mathcal{A}|}}{\arg\min}\quad & \sum_{a\in\mathcal{A}} \frac{\pi^2_{max}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]}{(1-\alpha)\pi_{log}(a|x)+\alpha\pi_x(a)}\\ \textit{subject to}\quad & \sum_{a\in\mathcal{A}} \pi_x(a) = 1,\\ & \pi_x(a) \geq 0 \textit{ for all } a \in \mathcal{A}\end{aligned}$$

This formulation also applies to the problem of augmentation logging for learning, since many algorithms involve some notion of an ambiguity class or trust region around the current learned policy π. OP3 allows one to apply MVAL whenever $\pi_{\max}(a|x)$ can be efficiently computed for these policy classes Π. We can even simplify OP3 for certain types of trust regions. For example, the trust region policy class $\Pi_{\pi_{\text{tar}}} = \{\pi | \forall x, a : \pi(a|x) \in [\frac{1}{\tau}\cdot\pi_{\text{tar}}(a|x), \tau\cdot \pi_{\text{tar}}(a|x)]\}$ can be approximated by setting $\pi_{\max} \approx \pi_{\text{tar}}$ for small $\tau \geq 1$. The argument is that the solution of OP3 is invariant to τ, since $\pi_{\max}(a|x) = \tau \cdot \pi_{\text{tar}}(a|x)$ as long as $\pi_{\text{tar}}(a|x) \leq \frac{1}{\tau}$.

4.1 Analysis and Discussion

We again illustrate and discuss the behavior of MVAL, now for the case of multi-policy evaluation and learning. An instructive limiting case is the situation where there is no past data, no restrictions on the policy class Π, and no knowledge about $\mathbb{E}_r\left[r^2(x,a)\right]$. In this case, it seems that one should sample from the uniform policy. We find this intuition agrees with MVAL for the case where Π is all valid policies, there is no logged data, and there is an uninformative reward model with $\mathbb{E}_r\left[r^2(x,a)\right] = c > 0$. This happens because when $\alpha = 1$, the optimal augmentation policy is

$$\frac{\pi_{\max}(a|x)\sqrt{\mathbb{E}_r\left[r^2(x,a)\right]}}{\sum_{a\in\mathcal{A}} \pi_{\max}(a|x)\sqrt{\mathbb{E}_r\left[r^2(x,a)\right]}}.$$

Since $\pi_{\max}(a|x) = \max_{\pi\in\Pi} \pi(a|x)$, then if Π is all valid policies then $\pi_{\max}(a|x) = 1$ for all a. Without a reward distribution model, $\mathbb{E}_r\left[r^2(x,a)\right] = c > 0$, so the optimal policy is the uniform policy:

$$\frac{1\sqrt{\mathbb{E}_r\left[r^2(x,a)\right]}}{\sum_{a\in\mathcal{A}} 1\sqrt{\mathbb{E}_r\left[r^2(x,a)\right]}} = \frac{\sqrt{c}}{\sum_{a\in\mathcal{A}}\sqrt{c}} = \frac{\sqrt{c}}{|\mathcal{A}|\sqrt{c}} = \frac{1}{|\mathcal{A}|}.$$

Therefore, if Π is the space of all valid policies, there is no existing logged data, and $\mathbb{E}_r\left[r^2(x,a)\right] = c > 0$ for all (x,a), then the optimal augmentation policy is the uniform distribution.

5 EMPIRICAL EVALUATION

To evaluate MVAL on a real-world contextual bandit problem, we performed experiments on the Yahoo! Front Page Dataset [8][2]. Each context in the dataset consists of a 5-dimensional vector $\mathbf{u}$ representing the user, as well as a $D \times 5$ dimensional matrix with vectors $\mathbf{A}_i$ for each of the D articles. For convenience, we only used contexts with exactly 19 articles. The dataset includes which article was recommended (i.e., the action) and if the user clicked on the article (i.e., the reward).

When collecting this dataset, the article recommendations were chosen uniformly at random from the articles available for the context. This allows for an unbiased simulation of running a different article recommendation policy π using rejection sampling [29]. To construct a sample for a new policy π, we iterate through the contexts x_i and sample $a' \sim \pi(\cdot|x)$ according to π. If the a' sampled from our policy agrees with the observed action a_i, we include that (x_i, a_i, r_i) tuple. This gives us an unbiased sample that comes from the same distribution as if we were running our new policy π on the operational system.

Model Architecture and Training. The policy architecture for precomputed MVAL is a feedforward neural network, ending in a softmax layer where the logit for article j is based on the concatenation of the user vector to the article vectors $\mathbf{u} \circ \mathbf{A}_j$ passed through 2 fully connected ReLU [20] layers with 256, 256 nodes before 1 fully connected linear node. Adam is used for optimization with the standard parameters $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ [15] and a batch size of 10000. All experiments can be run on a desktop with an RTX2080.

Estimating $\mathbb{E}_r\left[r^2(x,a)\right]$. The first set of experiments uses a uniform $\mathbb{E}_r\left[r^2(x,a)\right] = 1$, since these experiments explore how any decrease in variance is a result of the algorithmic improvement, rather than a result of a model of $\mathbb{E}_r\left[r^2(x,a)\right]$. For the sequential learning experiment, we use a feedforward neural network to approximate $\mathbb{E}\left[r^2(x,a)\right]$, trained using mean squared error to perform regression on the quantity $r(x,a)$ using the same architecture as the policy networks described above.

Generating Logging and Target Policies. The policy evaluation experiments use randomly generated target and logging policies. We control the generating process to vary how deterministic the logging policy's actions are, and how different the target and logging policies are. In particular, to generate the logging policy, we randomly sample a vector $\mathbf{v} \in \mathbb{R}^{25}$ such that $v_i \sim \mathcal{N}(\mu = 0, \sigma = 1)$. This vector is then multiplied against a feature vector of the cross terms between u and each A_i for the given context i to allow for interactions between user and article features while remaining a simple policy class. Then, the articles are ranked according to this value to define the selection probabilities of the policy based on the rank. In particular, the probability of choosing each article is proportional to η^{rank_i}, where rank_i is the rank of the ith article. This allows us to increase the determinism of the policy by increasing η. We use the same construction to generate a target policy, but shift a δ fraction of the probability weight from the top-ranked article under the logging policy to the article that is ranked second under the logging policy. Increasing δ allows us to increase the difference between the target and logging policy.

5.1 Single Policy Evaluation

The first experiments explore the effectiveness of MVAL for evaluating a single target policy while varying η and δ for the logging and target policy generation. The reported variance is the empirical variance of 50 sampled value estimates for the target policy. Each value estimate is generated by first sampling 900 data points from the logging policy. We then compare different augmentation logging policies that are allowed to sample 100 additional data points. Specifically, we compare MVAL and precomputed MVAL to using the target policy or the uniform policy for augmentation logging. Finally, we estimating the value of the target policy using the balanced estimator on all 1000 data points. The standard error bars are based on of 20 such runs for each parameter setting.

In Figure 1, we see that using MVAL substantially outperforms using the target policy or uniform policy for augmentation logging for a range determinism factors η. We find that precomputed MVAL performs comparable to MVAL over the whole range of η.

[2] This dataset was obtained from Yahoo! Webscope at http://research.yahoo.com/Academic_Relations. The users are anonymized.

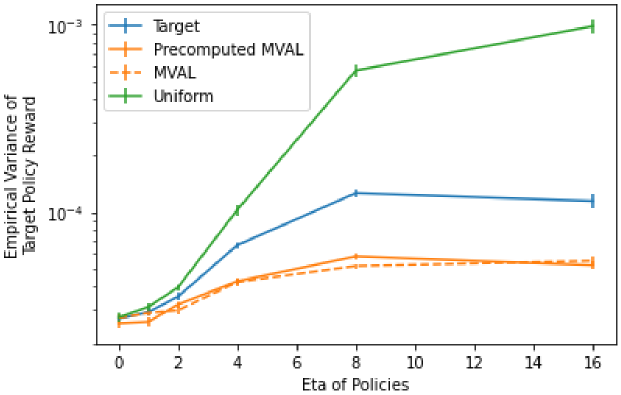

Figure 1: Variance of the balanced estimator while holding $\delta = 0.4$ and increasing values of η, which increases the determinism of the policies. Error bars are the standard error based on 20 trials. Note the logarithmic y axis.

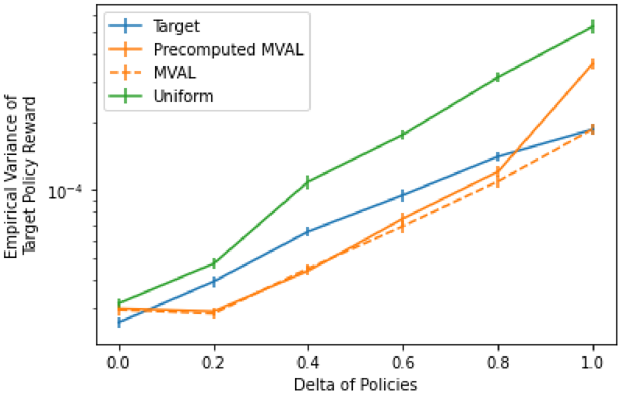

Figure 2: Variance of the balanced estimator while holding $\eta = 4$ and increasing values of δ, which increases the difference between the logging and target policies. Error bars are based on 20 trials.

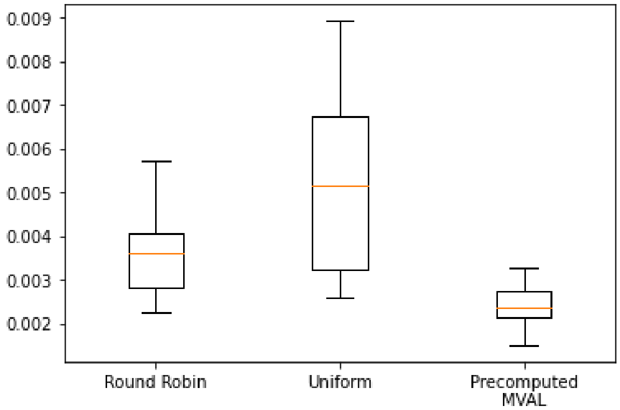

Figure 3: Variance of multiple policy evaluation for 3 policies with determinism factor $\eta = 4$ and change fraction $\delta = 0.4$. Each box plot contains all the empirical variances of all 3 policy estimates, for 20 trials of the full experiment.

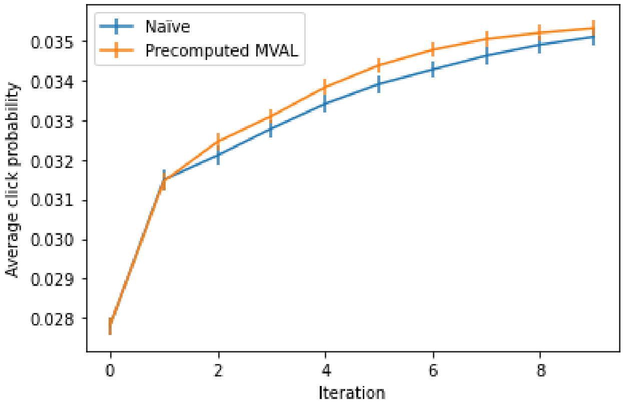

Figure 4: Average cumulative reward for sequentially using naïve or MVAL augmentation logging at each iteration for $\tau = 1$. Error bars are the standard error based on 50 trials.

The difference δ between logging and target policy is fixed at 0.4 in this experiment. At $\eta = 0$, the logging policy is the uniform distribution and, as expected, the advantage from augmentation logging is smallest.

In Figure 2, we vary the value of δ while the determinism factor η is fixed. Again, we find that using MVAL to generate the augmentation policy substantially outperforms using the target policy or uniform distribution for a broad range of settings. Increasing the change fraction δ decreases the performance of all methods as expected. At $\delta = 1$, MVAL and target tie because the target policy is fairly deterministic, and per the discussion in Section 3.2.4 the target policy is the variance-optimal augmentation policy. At $\delta = 1$, precomputed MVAL performs worse than exact MVAL, possibly because it is training a policy based on (x, a, r) tuples close to the logging policy rather than the quite different target policy.

5.2 Multiple Policy Evaluation

The next experiment explores the effectiveness of MVAL for multi-policy evaluation. The policies for this experiment were generated as before with determinism factor $\eta = 4$ and a difference between logging and target policy of $\delta = 0.4$. However, the 3 target policies were constructed by shifting δ fraction of the logging policy's top action's probability to the 2nd, 3rd, and 4th actions. The experiment variance is the empirical variance of 100 runs of using the balanced estimator to estimate the target policy reward. The round-robin strategy uses each target policy to sample 333 augmentation data points, while the other strategies collected 999 augmentation data points from the uniform or precomputed MVAL policy. The original logging policy provided 9001 data points for a total of 10000 contexts. As shown in Figure 3, precomputed MVAL substantially outperforms the round-robin and uniform strategies.

5.3 Policy Learning

The final experiment explores the effectivess of MVAL for policy learning over time, where we repeatedly gather 10000 additional augmentation data points and retrain the model using POEM [25] with the balanced estimator as policy learner. In the initial iteration we gather 10000 data points from the uniform policy, following the optimal MVAL strategy in this situation according to Section 4.1. After that, each iteration t trains a policy π^t using POEM [25] with the balanced estimator as policy learner. We compare using MVAL for augmentation logging to naively logging additional data from the target policy π^t in each iteration. We use precomputed MVAL with $\mathbb{E}_r\left[r^2(x,a)\right] = m^t(x,a)$, where m^t approximates $\mathbb{E}_r\left[r^2(x,a)\right]$ using regression. Furthermore, we approximate $\pi_{\max} \approx \pi^t$ as discussed in Section 4. For both the naive augmentation logging using π^t and MVAL augmentation logging we train POEM for 1000 epochs, which is more than sufficient for convergence. The POEM clipping parameter is set to 10000, which is chosen to maximize the performance of the naive method. Figure 4 shows that augmentation logging via MVAL outperforms naively using the target policy for augmentation logging.

6 CONCLUSIONS

We introduced and formalized the problem of augmentation logging, and derived MVAL as a principled and practical method for variance-optimizing data gathering for off-policy evaluation. We extended the approach to multi-policy evaluation and batch learning, and find that it can substantially improve estimation and learning quality over naive methods. This work opens up a number of directions for future work. For example, contextual-bandit problems with combinatorial actions like slates [27] raise additional challenges, but they also provide structure that connects the observations for different actions. It is interesting to explore how this structure can inform augmentation logging for improved bias/variance trade-offs.

ACKNOWLEDGMENTS

This research was supported in part by NSF Awards IIS-1901168 and IIS-2008139, and scholarship funding from Open Philanthropy. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

A APPENDIX

A.1 Variance of the Balanced Estimator

Since $\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$,

$$\mathrm{Var}\left[\frac{\pi_{\text{tar}}(a_i|x_i)}{\pi(a_i|x_i)} r_i\right] = \mathbb{E}\left[\left(\frac{\pi_{\text{tar}}(a_i|x_i)}{\pi(a_i|x_i)} r_i\right)^2\right] - \mathbb{E}\left[\frac{\pi_{\text{tar}}(a_i|x_i)}{\pi(a_i|x_i)} r_i\right]^2.$$

Since the IPS estimator is unbiased if the support of the logging policy $\pi(a|x)$ is a superset of $\pi_{\text{tar}}(a|x)$ for all x in the support of $\Pr(x)$, if we denote $\mathbb{E}_{\pi_{\text{tar}}}[r_i] = R_{\pi_{\text{tar}}}$, then

$$\mathrm{Var}\left[\frac{\pi_{\text{tar}}(a_i|x_i)}{\pi(a_i|x_i)} r_i\right] = \mathbb{E}\left[\left(\frac{\pi_{\text{tar}}(a_i|x_i)}{\pi(a_i|x_i)} r_i\right)^2\right] - R^2_{\pi_{\text{tar}}}.$$

Breaking up the expectations, we have the following:

$$\mathbb{E}\left[\left(\frac{\pi_{\text{tar}}(a_i|x_i)}{\pi(a_i|x_i)} r_i\right)^2\right] \tag{1}$$

$$= \mathbb{E}_{x\sim\Pr(x)}\,\mathbb{E}_{a\sim\pi(a|x)}\,\mathbb{E}_{r_i\sim r(x,a)}\left[\left(\frac{\pi_{\text{tar}}(a|x)}{\pi(a|x)} r_i\right)^2\right] \tag{2}$$

$$= \mathbb{E}_{x\sim\Pr(x)}\,\mathbb{E}_{a\sim\pi(a|x)}\left[\frac{\pi^2_{\text{tar}}(a|x)}{\pi^2(a|x)}\,\mathbb{E}_{r_i\sim r(x,a)}\left[r_i^2\right]\right] \tag{3}$$

$$= \mathbb{E}_{x\sim\Pr(x)}\left[\sum_{a\in\mathcal{A}} \frac{\pi^2_{\text{tar}}(a|x)}{\pi^2(a|x)}\,\mathbb{E}_{r_i\sim r(x,a)}\left[r_i^2\right]\pi(a|x)\right] \tag{4}$$

Equation 2 comes from iterated expectation, 3 from the fact that $\mathbb{E}_{r_i\sim r(x,a))}$ is conditioned on x and a (so they can be factored out), and 4 from the definition of $\mathbb{E}_{a\sim\pi(a|x)}$.

In the traditional IPS estimator, the two $\pi(a|x)$ terms cancel out. However, averaging $n_{\log}$ terms from $\pi_{\log}$ and n_{aug} terms from π_{aug} results in the following variance, where $N = n_{\log} + n_{\text{aug}}$:

$$\begin{aligned}\mathrm{Var}\left[\hat{R}^{\text{IPS}}_{\pi_{\text{tar}}}\right] = \frac{n_{\log}}{N^2}\,\mathbb{E}_x\left[\sum_{a\in\mathcal{A}} \frac{\pi^2_{\text{tar}}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]}{\pi_{\log}(a|x)} - R^2_{\pi_{\text{tar}}}\right] \\ + \frac{n_{\text{aug}}}{N^2}\,\mathbb{E}_x\left[\sum_{a\in\mathcal{A}} \frac{\pi^2_{\text{tar}}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]}{\pi_{\text{aug}}(a|x)} - R^2_{\pi_{\text{tar}}}\right]\end{aligned}$$

However, in this variance expression, the partial derivatives with respect to π_{aug} do not depend on any $\pi_{\log}$ terms, and so the evaluation policy cannot adapt to make up for any deficiencies in the logging policy. However, if instead we weighted terms by $\pi_{\text{tar}}(a|x)/\pi_{\text{balanced}}(a|x)$ as in the balanced estimator, we get the following variance:

$$\begin{aligned}&\mathrm{Var}\left[\hat{R}^{\text{BAL}}_{\pi_{\text{tar}}}\right] \\ &= \frac{n_{\log}}{N^2}\,\mathbb{E}_x\left[\sum_{a\in\mathcal{A}} \frac{\pi^2_{\text{tar}}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]}{\pi^2_{\text{balanced}}(a|x)}\pi_{\log}(a|x) - R^2_{\pi_{\text{tar}}}\right] \\ &+ \frac{n_{\text{aug}}}{N^2}\,\mathbb{E}_x\left[\sum_{a\in\mathcal{A}} \frac{\pi^2_{\text{tar}}(a|x)\,\mathbb{E}_r\left[r^2(x,a)\right]}{\pi^2_{\text{balanced}}(a|x)}\pi_{\text{aug}}(a|x) - R^2_{\pi_{\text{tar}}}\right]\end{aligned}$$

Since $\pi_{\text{balanced}}(a|x) = \frac{n_{\log}}{N}\pi_{\log}(a|x) + \frac{n_{\text{aug}}}{N}\pi_{\text{aug}}(a|x)$, we have the following final variance:

$$\mathrm{Var}\left[\hat{R}^{\text{BAL}}_{\pi_{\text{tar}}}\right] = \frac{1}{N}\left[\mathbb{E}_x\left[\sum_{a\in\mathcal{A}} \frac{\pi^2_{\text{tar}}(a|x)}{\pi_{\text{balanced}}(a|x)}\,\mathbb{E}_r\left[r^2(x,a)\right]\right] - R^2_{\pi_{\text{tar}}}\right].$$

An observant reader may worry that these variance terms are not independent if the augmentation policy π_{aug} depends on the logging policy $\pi_{\log}$. However, note that if π_{aug} is chosen to minimize this variance, while π_{aug} depends on $\pi_{\log}$ and π_{tar} and the distribution over contexts $\mathcal{D}$, it does not depend on any realization of the random variables. Therefore while π_{aug} depends on $\pi_{\log}$ and $\mathcal{D}$, the individual IPS terms are independent after conditioning on $\pi_{\log}$, π_{aug}, $\mathcal{D}$, and other known terms.

REFERENCES

[1] A. Agarwal, S. Basu, T. Schnabel, and T. Joachims. 2017. Effective Evaluation using Logged Bandit Feedback from Multiple Loggers. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

[2] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. 2014. Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits. In *Proceedings of the 31st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 32)*, Eric P. Xing and Tony Jebara (Eds.). PMLR, Bejing, China, 1638–1646. http://proceedings.mlr.press/v32/agarwalb14.html

[3] Shipra Agrawal and Navin Goyal. 2013. Thompson Sampling for Contextual Bandits with Linear Payoffs. In *Proceedings of the 30th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 28)*, Sanjoy Dasgupta and David McAllester (Eds.). PMLR, Atlanta, Georgia, USA, 127–135. http://proceedings.mlr.press/v28/agrawal13.html

[4] Alina Beygelzimer and John Langford. 2009. The offset tree for learning with partial labels. In *KDD*. ACM, 129–138.

[5] Alberto Bietti, Alekh Agarwal, and John Langford. 2018. A contextual bandit bake-off. *arXiv preprint arXiv:1802.04064* (2018).

[6] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising. *Journal of Machine Learning Research* 14, 65 (2013), 3207–3260. http://jmlr.org/papers/v14/bottou13a.html

[7] N. Cesa-Bianchi and G. Lugosi. 2006. *Prediction, Learning, and Games*. Cambridge University Press.

[8] Wei Chu, Seung-Taek Park, Todd Beaupre, Nitin Motgi, Amit Phadke, Seinjuti Chakraborty, and Joe Zachariah. 2009. A case study of behavior-driven conjoint analysis on Yahoo! Front Page Today module. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1097–1104.

[9] JEAN-MARIE CORNUET, JEAN-MICHEL MARIN, Antonietta Mira, and Christian P Robert. 2012. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics* 39, 4 (2012), 798–812.

[10] 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*. 1097–1104.

[11] Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. 2018. More Robust Doubly Robust Off-policy Evaluation. In *ICML*.

[12] Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. 2019. Batched Multi-armed Bandits Problem. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/20f07591c6fcb220ffe637cda29bb3f6-Paper.pdf

[13] Daniel G Horvitz and Donovan J Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association* 47, 260 (1952), 663–685.

[14] T. Joachims, A. Swaminathan, and M. de Rijke. 2018. Deep Learning with Logged Bandit Feedback. In *ICLR*.

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] John Langford and Tong Zhang. 2007. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in neural information processing systems* 20 (2007), 817–824.

[17] Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. 2015. Counterfactual estimation and optimization of click metrics in search engines: A case study. In *Proceedings of the 24th International Conference on World Wide Web*. 929–934.

[18] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. 661–670.

[19] Ben London and Ted Sandler. 2019. Bayesian Counterfactual Risk Minimization. In *ICML*. 4125–4133.

[20] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*.

[21] Art Owen. 2013. *Monte Carlo Theory, Methods and Examples*. Stanford.

[22] N. Sachdeva, Yi Su, and T. Joachims. 2020. Off-policy Bandits with Deficient Support. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

[23] Alex Strehl, John Langford, Lihong Li, and Sham M Kakade. 2011. Learning from Logged Implicit Exploration Data. In *NeurIPS*.

[24] Yi Su, Lequn Wang, Michele Santacatterina, and Thorsten Joachims. 2019. CAB: Continuous Adaptive Blending for Policy Evaluation and Learning. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 6005–6014. http://proceedings.mlr.press/v97/su19a.html

[25] Adith Swaminathan and Thorsten Joachims. 2015. Batch Learning from Logged Bandit Feedback through Counterfactual Risk Minimization. *Journal of Machine Learning Research* 16, 52 (2015), 1731–1755. http://jmlr.org/papers/v16/swaminathan15a.html

[26] A. Swaminathan and T. Joachims. 2015. The Self-Normalized Estimator for Counterfactual Learning. In *NeurIPS*.

[27] A. Swaminathan, A. Krishnamurthy, A. Agarwal, M. Dudik, J. Langford, D. Jose, and I. Zitouni. 2017. Off-policy Evaluation for Slate Recommendation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[28] Philip Thomas and Emma Brunskill. 2016. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2139–2148.

[29] Hastagiri P. Vanchinathan, Isidor Nikolic, Fabio De Bona, and Andreas Krause. 2014. Explore-Exploit in Top-N Recommender Systems via Gaussian Processes. In *Proceedings of the 8th ACM Conference on Recommender Systems* (Foster City, Silicon Valley, California, USA) *(RecSys '14)*. Association for Computing Machinery, New York, NY, USA, 225–232. https://doi.org/10.1145/2645710.2645733

[30] Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudík. 2017. Optimal and Adaptive Off-policy Evaluation in Contextual Bandits. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, International Convention Centre, Sydney, Australia, 3589–3597. http://proceedings.mlr.press/v70/wang17a.html

[31] Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. 2018. Active Learning with Logged Data. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 5521–5530. https://proceedings.mlr.press/v80/yan18a.html

[32] Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. 2019. The Label Complexity of Active Learning from Observational Data. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/1019c8091693ef5c5f55970346633f92-Paper.pdf