Effective Evaluation Using Logged Bandit Feedback from Multiple Loggers

Aman Agarwal, Soumya Basu, Tobias Schnabel, Thorsten Joachims
Cornell University, Dept. of Computer Science
Ithaca, NY, USA
[aa2398,sb2352,tbs49,tj36]@cornell.edu

ABSTRACT

Accurately evaluating new policies (e.g. ad-placement models, ranking functions, recommendation functions) is one of the key prerequisites for improving interactive systems. While the conventional approach to evaluation relies on online A/B tests, recent work has shown that counterfactual estimators can provide an inexpensive and fast alternative, since they can be applied offline using log data that was collected from a different policy fielded in the past. In this paper, we address the question of how to estimate the performance of a new target policy when we have log data from multiple historic policies. This question is of great relevance in practice, since policies get updated frequently in most online systems. We show that naively combining data from multiple logging policies can be highly suboptimal. In particular, we find that the standard Inverse Propensity Score (IPS) estimator suffers especially when logging and target policies diverge – to a point where throwing away data improves the variance of the estimator. We therefore propose two alternative estimators which we characterize theoretically and compare experimentally. We find that the new estimators can provide substantially improved estimation accuracy.

CCS CONCEPTS

•**Computing methodologies** →**Learning from implicit feedback;** *Causal reasoning and diagnostics;* •**Information systems** →*Evaluation of retrieval results;*

KEYWORDS

counterfactual estimators, log data, implicit feedback, off-policy evaluation

1 INTRODUCTION

Interactive systems (e.g., search engines, ad-placement systems, recommender systems, e-commerce sites) are typically evaluated according to online metrics (e.g., click through rates, dwell times) that reflect the users' response to the actions taken by the system. For this reason, A/B tests are of widespread use in which the new policy to be evaluated is fielded to a subsample of the user population. Unfortunately, A/B tests come with two drawbacks. First, they can be detrimental to the user experience if the new policy to be evaluated performs poorly. Second, the number of new policies that can be evaluated in a given amount of time is limited, simply because each A/B test needs to be run on a certain fraction of the overall traffic and should ideally span any cycles (e.g. weekly patterns) in user behavior.

Recent work on counterfactual evaluation techniques provides a principled alternative to A/B tests that does not have these drawbacks [2, 11, 13, 21]. These techniques do not require that the new policy be deployed online, but they instead allow reusing logged interaction data that was collected by a different policy in the past. In this way, these estimators address the counterfactual inference question of how a new policy would have performed, if it had been deployed instead of the old policy that actually logged the data. This allows reusing the same logged data for evaluating many new policies, greatly improving scalability and timeliness compared to A/B tests.

In this paper, we address the problem of counterfactual evaluation when log data is available not just from one logging policy, but from multiple logging policies. Having data from multiple policies is common to most practical settings where systems are repeatedly modified and deployed. While the standard counterfactual estimators based on inverse propensity scores (IPS) apply to this situation, we show that they are suboptimal in terms of their estimation quality. In particular, we investigate the common setting where the log data takes the form of contextual bandit feedback from a stochastic policy, showing that the variance of the conventional IPS estimator suffers substantially when the historic policies are sufficiently different – to a point where throwing away data improves the variance of the estimator. To overcome the statistical inefficiency of the conventional IPS estimator, we explore two alternative estimators that directly account for the data coming from multiple different logging policies. We show theoretically that both estimators are unbiased, and have lower variance than the conventional IPS estimator. Furthermore, we quantify the amount of variance reduction in an extensive empirical evaluation that demonstrates the effectiveness of both the estimators.

2 RELATED WORK

The problem of re-using logged bandit feedback is often part of counterfactual learning [2, 11, 21], and more generally can be viewed as part of off-policy evaluation in reinforcement learning [17, 20].

In counterfactual learning, solving the evaluation problem is often the first step to deriving a learning algorithm [2, 19, 21]. The key to being able to counterfactually reason based on logged

KDD'17, August 13–17, 2017, Halifax, NS, Canada.

978-1-4503-4887-4/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3097983.3098155

data is randomness in the logged data. Approaches differ in how randomness is being included in the policies. For example, in [11] randomization is directly applied to the actions of each policy, whereas [2] randomizes individual policy parameters to create a distribution over actions.

In exploration scavenging [10], the authors address counterfactual evaluation in a setting where the actions do not depend on the context. They mention the possibility of combining data from different policies by interpreting each policy as an action. Li et al. [14] propose to use naturally occurring randomness in the logged data when policies change due to system changes. Since this natural randomness may not be entirely under the operator's control, the authors propose to estimate the probability that a certain logging policy was in place to recover propensities. The balanced IPS estimator studied in this paper could serve as a starting point for further techniques in that direction.

Evaluation from logged data has often been studied with respect to specific domains, for example in news recommendation [11–13] as well as in information retrieval [8, 11]. The work by Li et al. [13] highlights another common use-case in practice, where different logging policies are all active at the same time, focusing on the evaluation of different new methods. The estimators in this paper can naturally be applied to this scenario as well to augment logging data of one policy with the data from others. An interesting example for probabilistic policies can be found in [8], where the authors consider policies that are the probabilistic interleaving of two deterministic ranking policies and use log data to pre-select new candidate policies.

Very related to combining logs from different policies is the problem of combining samples coming from different proposal distributions in importance sampling [5, 15, 16]. There, samples are drawn from multiple proposal distributions and need to be combined in a way that reduces variance of the combined estimator. Multiple importance sampling has been particularly studied in computer graphics [22], as Monte Carlo techniques are employed for rendering. Most related to the weighted IPS estimator presented later in the paper is adaptive multiple importance sampling (AMIS) [4, 6] that also recognizes that it is not optimal to weigh contributions from all proposal distributions the same, but instead updates weights as well as the proposal distributions after each sampling step. The most notable differences to our setting here are that (i) we regard the sampling distributions as given and fixed, and (ii) the sampled log data is also fixed. An interesting avenue for future work would be to use control variates to further reduce variance of our estimators [7, 15], although this approach is computationally demanding since it requires solving a quadratic problem to determine optimal weights.

Another related area is sampling-based evaluation of information retrieval systems [3, 18, 23]. Instead of feedback data that stems from interactions with users, the observed feedback comes from judges. A policy in this case corresponds to a sampling strategy which determines the query-document pairs to be sent out for judgement. As shown by Carterette et al. [3], relying on sampling-based elicitation schemes cuts down the number of required judgements substantially as compared to a classic deterministic pooling scheme. The techniques proposed in our paper could also be applied to the evaluation of retrieval systems when data from different judgement pools need to be combined.

3 PROBLEM SETTING

In this paper, we study the use of logged Bandit feedback that arises in interactive learning systems. In these systems, the system receives as input a vector $x \in \mathcal{X}$, typically encoding user input or other contextual information. Based on input x, the system responds with an action $y \in \mathcal{Y}$ for which it receives some feedback in the form of a cardinal utility value $\delta : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$. Since the system only receives feedback for the action y that it actually takes, this feedback is often referred to as Bandit feedback [21].

For example, in ad placement models, the input x typically encodes user-specific information as well as the web page content, and the system responds with an ad y which is then displayed on the page. Finally, user feedback $\delta(x, y)$ for the displayed ad is presented, such as whether the ad was clicked or not. Similarly, for a news website, the input x may encode user-specific and other contextual information to which the system responds with a personalized home page y. In this setting, the user feedback $\delta(x, y)$ could be the time spent by the user on the news website.

In order to be able to counterfactually evaluate new policies, we consider *stochastic policies* π that define a probability distribution over the output space $\mathcal{Y}$. Predictions are made by sampling $y \sim \pi(\mathcal{Y}|x)$ from a policy given input x. The inputs are assumed to be drawn i.i.d. from a fixed but unknown distribution $x \overset{i.i.d.}{\sim} Pr(\mathcal{X})$. The feedback $\delta(x, y)$ is a cardinal utility that is only observed at the sampled data points. Large values for $\delta(x, y)$ indicate user satisfaction with y for x, while small values indicate dissatisfaction.

We evaluate and compare different policies with respect to their induced utilities. The utility of a policy $U(\pi)$ is defined as the expected utility of its predictions under both the input distribution as well as the stochastic policy. More formally:

Definition 3.1 (Utility of Policy). The utility of a policy π is

$$\begin{aligned} U(\pi) &\equiv \mathbb{E}_{x\sim\Pr(\mathcal{X})}\mathbb{E}_{y\sim\pi(\mathcal{Y}|x)}[\delta(x, y)] \\ &= \sum_{\substack{x\in\mathcal{X} \\ y\in\mathcal{Y}}} \Pr(x)\pi(y|x)\delta(x, y) \end{aligned}$$

Our goal is to re-use the interaction logs collected from multiple historic policies to estimate the utility of a new policy. In this paper, we denote the the new policy (also called the target policy) as $\bar{\pi}$, and the m logging policies as $\pi_1, \ldots, \pi_n$. The log data collected from each logging policy π_i is

$$\mathcal{D}^i = \{(x_1^i, y_1^i, \delta_1^i, p_1^i), \ldots, (x_{n_i}^i, y_{n_i}^i, \delta_{n_i}^i, p_{n_i}^i)\},$$

where n_i data-points are collected from logging policy π_i, $x_j^i \sim \Pr(\mathcal{X})$, $y_j^i \sim \pi_i(\mathcal{Y}|x_j^i)$, $\delta_j^i \equiv \delta(x_j^i, y_j^i)$, and $p_j^i \equiv \pi_i(y_j^i|x_j^i)$. Note that during the operation of the logging policies, the propensities $\pi_i(y|x)$ are tracked and appended to the logs. We will also assume that the quantity $\pi_i(y|x)$ is available at all (x, y) pairs. This is a very mild assumption since the logging policies were designed and controlled by us, so their code can be stored. Finally, let $\mathcal{D} = \bigcup_{i=1}^{m} \mathcal{D}^i$ denote the combined collection of log data over all the logging policies, and $n = \sum_{i=1}^{m} n_i$ denote the total number of samples.

Unfortunately, it is not possible to directly compute the utility of a policy based on log data using the formula from the definition above. While we have a random sample of the contexts x and the target policy $\pi(y|x)$ is known by construction, we lack full information about the feedback $\delta(x, y)$. In particular, we know $\delta(x, y)$ only for the particular action chosen by the logging policy, but we do not necessarily know it for all the actions that the target policy $\pi(y|x)$ can choose. In short, we only have logged bandit feedback, but not full-information feedback. This motivates the use of statistical estimators to overcome the infeasibility of exact computation. In the following sections, we will explore three such estimators and focus on two of their key statistics properties, namely their bias and variance.

4 NAIVE INVERSE PROPENSITY SCORING

A natural first candidate to explore for the evaluation problem using multiple logging policies as defined above is the well-known inverse propensity score (IPS) estimator. It simply averages over all datapoints, and corrects for the distribution mismatch betweenthe logging policies π_i and the target policy $\bar{\pi}$ using a weighting term:

Definition 4.1 (Naive IPS Estimator).

$$\hat{U}_{naive}(\bar{\pi}) \equiv \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n_i}\delta_j^i \frac{\bar{\pi}(y_j^i|x_j^i)}{p_j^i}.$$

This is an unbiased estimator as shown below, as long as all logging policies have full support for the new policy $\bar{\pi}$.

Definition 4.2 (Support). Policy π is said to have *support* for policy π' if for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$,

$$\delta(x, y)\pi'(y|x) \neq 0 \Rightarrow \pi(y|x) > 0.$$

Proposition 4.3 (Bias of Naive IPS Estimator). *Assume each logging policy π_i has support for target $\bar{\pi}$. For $\mathcal{D}$ consisting of i.i.d. draws from* $\Pr(\mathcal{X})$ *and logging policies $\pi_i(\mathcal{Y}|x)$, the naive IPS estimator is unbiased:*

$$\mathbb{E}_{\mathcal{D}}[\hat{U}_{naive}(\bar{\pi})] = U(\bar{\pi}).$$

Proof. By linearity of expectation,

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}[\hat{U}_{naive}(\bar{\pi})] &= \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n_i}\mathbb{E}_{x\sim\Pr(\mathcal{X}),y\sim\pi_i(\mathcal{Y}|x)}\left[\frac{\delta(x,y)\bar{\pi}(y|x)}{\pi_i(y|x)}\right] \\
&= \frac{1}{n}\sum_{i=1}^{m} n_i \sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\Pr(x)\pi_i(y|x)\frac{\delta(x,y)\bar{\pi}(y|x)}{\pi_i(y|x)} \\
&= \sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\Pr(x)\delta(x,y)\bar{\pi}(y|x) \\
&= \mathbb{E}_{x\sim Pr(\mathcal{X})}\mathbb{E}_{y\sim\bar{\pi}(\mathcal{Y}|x)}[\delta(x,y)] \\
&= U(\bar{\pi}).
\end{aligned}$$

The second equality is valid since each π_i has support for $\bar{\pi}$. □

Note that the requirement that the logging policies π_i have support for the target policy can be satisfied by ensuring that $\pi_i(y|x) > \epsilon$ when deploying policies.

		x_1	x_2
Pr(x)		0.5	0.5
$\delta(x, y)$	y_1	10	1
	y_2	1	10
$\pi_1(y\|x)$	y_1	0.2	0.8
	y_2	0.8	0.2
$\pi_2(y\|x)$	y_1	0.9	0.1
	y_2	0.1	0.9
$\bar{\pi}(y\|x)$	y_1	0.8	0.2
	y_2	0.2	0.8

Table 1: Dropping data samples from logging policy π_1 lowers the variance of the naive and balanced IPS estimators when estimating the utility of $\bar{\pi}$.

We can also characterize the variance of the naive IPS estimator.

$$\begin{aligned}
&\mathrm{Var}_{\mathcal{D}}[\hat{U}_{naive}(\bar{\pi})] \qquad (1)\\
&= \frac{1}{n^2}\sum_{i=1}^{m} n_i \left(\sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\frac{(\delta(x,y)\bar{\pi}(y|x))^2}{\pi_i(y|x)}\Pr(x) - U(\bar{\pi})^2\right).
\end{aligned}$$

Having characterized both the bias and the variance of the Naive IPS Estimator, how does it perform on datasets that come from multiple logging policies?

4.1 Suboptimality of Naive IPS Estimator

To illustrate the suboptimality of the Naive IPS Estimator when we have data from multiple logging policies, consider the following toy example where we wish to evaluate a new policy $\bar{\pi}$ given data from two logging policies π_1 and π_2. For simplicity and without loss of generality, consider logged bandit feedback which consists of one sample from π_1 and another sample from π_2, more specifically, we have two logs $\mathcal{D}^1 = \{(x_1^1, y_1^1, \delta_1^1, p_1^1)\}$, and $\mathcal{D}^2 = \{(x_1^2, y_1^2, \delta_1^2, p_1^2)\}$. There are two possible inputs x_1, x_2 and two possible output predictions y_1, y_2. The cardinal utility function δ, the input distribution $\Pr(\mathcal{X})$, the target policy $\bar{\pi}$, and the two logging policies π_1 and π_2 are given in Table 1.

From the table, we can see that the target policy $\bar{\pi}$ is similar to logging policy π_2, but that it is substantially different from π_1. Since the mismatch between target and logging policy enters the IPS estimator as a ratio, one would like to keep that ratio small for low variance. That, intuitively speaking, means that samples from π_2 result in lower variance than samples from π_1, and that the π_1 samples may be adding a large amount of variability to the estimate. Indeed, it turns out that simply omitting the data from $\mathcal{D}^1$ greatly improves the variance of the estimator. Plugging the appropriate values into the variance formula in Equation (1) shows that the variance $\mathrm{Var}_{\mathcal{D}}[\hat{U}_{naive}(\bar{\pi})]$ is reduced from 64.27 to 4.27 by dropping the sample from the first logging policy π_1. Intuitively, the variance of $\hat{U}_{naive}(\bar{\pi})$ suffers because higher variance samples from one logging policy drown out the signal from the lower variance samples to an extent that can even dominate the benefit of having more samples. Thus, $\hat{U}_{naive}(\bar{\pi})$ fails to make the most of the available log data by combining it in an overly naive way.

Under closer inspection of Equation (1), the fact that deleting data helps improve variance also makes intuitive sense. Since the overall variance contains the sum of variances over all individual samples, one can hope to improve variance by leaving out high-variance samples. This motivates the estimators we introduce in the following sections, and we will show how weighting samples generalizes this variance-minimization strategy.

5 ESTIMATOR FROM MULTIPLE IMPORTANCE SAMPLING

Having seen that $\hat{U}_{naive}(\bar{\pi})$ has suboptimal variance, we first explore an alternative estimator used in multiple importance sampling [16]. We begin with a brief review of multiple importance sampling.

Suppose there is a target distribution p on $\mathcal{S} \subseteq \mathbb{R}^d$, a function f, and $\mu = \mathbb{E}_p(f(\mathbf{X})) = \int_{\mathcal{S}} f(x)p(x)\mathrm{d}x$ is the quantity to be estimated. The function f is observed only at the sampled points. In multiple importance sampling, n_j observations $x_{ij} \sim \mathbf{X}$, $i \in [n_j]$ are taken from sampling distributions q_j for $j = 1, \dots, J$. An unbiased estimator that is known to have low variance in this case is the *balance heuristic* estimate [16];

$$\tilde{\mu}_\alpha = \frac{1}{n}\sum_{j=1}^{J}\sum_{i=1}^{n_j}\frac{f(x_{ij})p(x_{ij})}{\sum_{j=1}^{J}\alpha_j q_j(x_{ij})},$$

where $n = \sum_{j=1}^{J} n_j$, and $\alpha_j = \frac{n_j}{n}$. Directly mapping the above to our setting, we define the Balanced IPS Estimator as follows.

Definition 5.1 (Balanced IPS Estimator).

$$\hat{U}_{bal}(\bar{\pi}) = \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n_i}\delta_j^i\frac{\bar{\pi}(y_j^i|x_j^i)}{\pi_{avg}(y_j^i|x_j^i)},$$

where for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, $\pi_{avg}(y|x) = \frac{\sum_{i=1}^{m} n_i \pi_i(y|x)}{n}$.

Note that π_{avg} is a valid policy since the convex combination of probability distributions is a probability distribution. The balanced IPS estimator $\hat{U}_{bal}(\bar{\pi})$ is also unbiased. Note that it now suffices that π_{avg} has support, but not necessarily that each individual π_i has support.

Proposition 5.2 (Bias of Balanced IPS Estimator). *Assume the policy π_{avg} has support for target $\bar{\pi}$. For $\mathcal{D}$ consisting of i.i.d. draws from* $\Pr(\mathcal{X})$ *and logging policies $\pi_i(\mathcal{Y}|x)$, the Balanced IPS Estimator is unbiased:*

$$\mathbb{E}_{\mathcal{D}}[\hat{U}_{bal}(\bar{\pi})] = U(\bar{\pi}).$$

Proof. By linearity of expectation,

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}[\hat{U}_{bal}(\bar{\pi})] &= \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n_i}\mathbb{E}_{x\sim\Pr(\mathcal{X}),\, y\sim\pi_i(\mathcal{Y}|x)}\left[\frac{\delta(x,y)\bar{\pi}(y|x)}{\pi_{avg}(y|x)}\right]\\
&= \frac{1}{n}\sum_{i=1}^{m} n_i \sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\Pr(x)\pi_i(y|x)\frac{\delta(x,y)\bar{\pi}(y|x)}{\pi_{avg}(y|x)}\\
&= \frac{1}{n}\sum_{x\in\mathcal{X}, y\in\mathcal{Y}}\frac{\Pr(x)\delta(x,y)\bar{\pi}(y|x)}{\pi_{avg}(y|x)}\sum_{i=1}^{m} n_i\pi_i(y|x)\\
&= \frac{1}{n}\sum_{x\in\mathcal{X}, y\in\mathcal{Y}}\frac{\Pr(x)\delta(x,y)\bar{\pi}(y|x)}{\frac{\sum_{i=1}^{m} n_i\pi_i(y|x)}{n}}\sum_{i=1}^{m} n_i\pi_i(y|x)\\
&= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}}\Pr(x)\delta(x,y)\bar{\pi}(y|x)\\
&= \mathbb{E}_{x\sim Pr(\mathcal{X})}\mathbb{E}_{y\sim\bar{\pi}(\mathcal{Y}|x)}[\delta(x,y)]\\
&= U(\bar{\pi}).
\end{aligned}$$

The second equality is valid since π_{avg} has support for $\bar{\pi}$. □

The variance of $\hat{U}_{bal}(\bar{\pi})$ can be computed as follows:

$$\begin{aligned}
\mathrm{Var}_{\mathcal{D}}[\hat{U}_{bal}(\bar{\pi})] = \frac{1}{n^2}\sum_{i=1}^{m} n_i\Bigg(&\sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\frac{(\delta(x,y)\bar{\pi}(y|x))^2}{\pi_{avg}(y|x)^2}\pi_i(y|x)\Pr(x)\\
&-\left(\sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\frac{(\delta(x,y)\bar{\pi}(y|x))}{\pi_{avg}(y|x)}\pi_i(y|x)\Pr(x)\right)^2\Bigg).
\end{aligned}$$

A direct consequence of Theorem 1 in [22] is that the variance of the balanced estimator is bounded above by the variance of the naive estimator plus some positive term that depends on $U(\bar{\pi})$ and the log sizes n_i.

Here, we provide a stronger result that does not require an extra positive term for the inequality to hold.

Theorem 5.3. *Assume each logging policy π_i has support for target $\bar{\pi}$. We then have that*

$$\mathrm{Var}_{\mathcal{D}}[\hat{U}_{bal}(\bar{\pi})] \leq \mathrm{Var}_{\mathcal{D}}[\hat{U}_{naive}(\bar{\pi})].$$

Proof. From Equation 1, we have the following expression.

$$\begin{aligned}
&\mathrm{Var}_{\mathcal{D}}[\hat{U}_{naive}(\bar{\pi})]\\
&\quad= \frac{1}{n^2}\sum_{i=1}^{m} n_i\left(\sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\frac{(\delta(x,y)\bar{\pi}(y|x))^2}{\pi_i(y|x)}\Pr(x) - U(\bar{\pi})^2\right).
\end{aligned}$$

For convenience, and without loss of generality, assume $n_i = 1$ $\forall i$, and therefore, $n = m$. This is easily achieved by re-labeling the logging policies so that each data-sample comes from a distinctly labeled policy (note that we don't need the logging policies to be distinct in our setup). Also, for simplicity, let $c(x, y) = \delta(x, y)\bar{\pi}(y|x)$. Then

$$
\begin{aligned}
&\mathrm{Var}_{\mathcal{D}}[\hat{U}_{naive}(\bar{\pi})] \geq \mathrm{Var}_{\mathcal{D}}[\hat{U}_{bal}(\bar{\pi})] \\
&\Leftrightarrow \sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}} c^2(x,y)\Pr(x)\Big(\sum_{i=1}^{m}\frac{1}{\pi_i(y|x)}\Big) - mU(\bar{\pi})^2 \\
&\geq \sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}} \frac{c^2(x,y)\Pr(x)}{\pi_{avg}(y|x)^2}\left(\sum_{i=1}^{m}\pi_i(y|x)\right) - \sum_{i=1}^{m}\left(\sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\frac{c(x,y)\Pr(x)}{\pi_{avg}(y|x)}\pi_i(y|x)\right)^2
\end{aligned}
$$

Thus, it is sufficient to show the following two inequalities

$$
\sum_{i=1}^{m}\left(\sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\frac{c(x,y)\Pr(x)}{\pi_{avg}(y|x)}\pi_i(y|x)\right)^2 \geq mU(\bar{\pi})^2 \tag{2}
$$

and for all relevant x, y

$$
\sum_{i=1}^{m}\frac{1}{\pi_i(y|x)} \geq \frac{1}{\pi_{avg}(y|x)^2}\left(\sum_{i=1}^{m}\pi_i(y|x)\right) \tag{3}
$$

We get Equation 2 by applying Cauchy-Schwarz as follows

$$
\begin{aligned}
&\left(\sum_{i=1}^{m}1^2\right)\left(\sum_{i=1}^{m}\left(\sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\frac{c(x,y)\Pr(x)}{\pi_{avg}(y|x)}\pi_i(y|x)\right)^2\right) \\
&\geq \left(\sum_{i=1}^{m}\sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\frac{c(x,y)\Pr(x)}{\pi_{avg}(y|x)}\pi_i(y|x)\right)^2 \\
&\Rightarrow \sum_{i=1}^{m}\left(\sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\frac{c(x,y)\Pr(x)}{\pi_{avg}(y|x)}\pi_i(y|x)\right)^2 \\
&\geq \left(\sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\frac{c(x,y)\Pr(x)}{\frac{1}{m}\sum_{i=1}^{m}\pi_i(y|x)}\sum_{i=1}^{m}\pi_i(y|x)\right)^2 = mU(\bar{\pi})^2
\end{aligned}
$$

Another application of Cauchy-Schwarz gives us Equation 3 in the following way

$$
\begin{aligned}
&\left(\sum_{i=1}^{m}\frac{1}{\pi_i(y|x)}\right)\left(\sum_{i=1}^{m}\pi_i(y|x)\right) \geq m^2 \\
&\Rightarrow \sum_{i=1}^{m}\frac{1}{\pi_i(y|x)} \geq \frac{1}{(\frac{1}{m}\sum_{i=1}^{m}\pi_i(y|x))^2}\sum_{i=1}^{m}\pi_i(y|x) \\
&= \frac{1}{\pi_{avg}(y|x)^2}\left(\sum_{i=1}^{m}\pi_i(y|x)\right)
\end{aligned}
$$

□

Returning to our toy example in Table 1, we can check the variance reduction provided by $\hat{U}_{bal}(\bar{\pi})$ over $\hat{U}_{naive}(\bar{\pi})$. The variance of the Balanced IPS Estimator is $\mathrm{Var}_{\mathcal{D}}[\hat{U}_{bal}(\bar{\pi})] \approx 12.43$, which is substantially smaller than $\mathrm{Var}_{\mathcal{D}}[\hat{U}_{naive}(\bar{\pi})] \approx 64.27$ for the naive estimator using all the data $\mathcal{D} = \mathcal{D}^1 \bigcup \mathcal{D}^2$. However, the Balanced IPS Estimator still improves when removing $\mathcal{D}^1$. In particular, notice that when using only $\mathcal{D}^2$, the variance of the Balanced IPS Estimator is $\mathrm{Var}_{\mathcal{D}}[\hat{U}_{bal}(\bar{\pi})] = \mathrm{Var}_{\mathcal{D}}[\hat{U}_{naive}(\bar{\pi})] \approx 4.27 < 12.43$. Therefore, even the variance of $\hat{U}_{bal}(\bar{\pi})$ can be improved in some cases by dropping data.

6 WEIGHTED IPS ESTIMATOR

We have seen that the variances of both the Naive and the Balanced IPS estimators can be reduced by removing some of the data points. More generally, we now explore estimators that re-weight samples from various logging policies based on their relationship with the target policy. This is similar to ideas that are used in Adaptive Multiple Importance Sampling [4, 6] where samples are also re-weighted in each sampling round. In contrast to the latter scenario, here we assume the logging policies to be fixed, and we derive closed-form formulas for variance-optimal estimators. The general idea of the weighted estimators that follow is to compute a weight for each logging policy that captures the mismatch between this policy and the target policy. In order to characterize the relationship between a logging policy and the new policy to be evaluated, we define the following *divergence*. This formalizes the notion of mismatch between the two policies in terms of the Naive IPS Estimator variance.

Definition 6.1 (Divergence). Suppose policy π has support for target policy $\bar{\pi}$. Then the divergence from π to $\bar{\pi}$ is

$$
\begin{aligned}
\sigma_{\delta}^2(\bar{\pi}||\pi) &\equiv \mathrm{Var}_{x\sim\Pr(\mathcal{X}),y\sim\pi(\mathcal{Y}|x)}\left[\frac{\delta(x,y)\bar{\pi}(y|x)}{\pi(y|x)}\right] \\
&= \sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}}\frac{(\delta(x,y)\bar{\pi}(y|x))^2}{\pi(y|x)}\Pr(x) - U(\bar{\pi})^2.
\end{aligned}
$$

Recall that $U(\bar{\pi})$ is the utility of policy $\bar{\pi}$.

Note that $\sigma_{\delta}^2(\bar{\pi}||\pi)$ is not necessarily minimal when $\pi = \bar{\pi}$. In fact, it can easily be seen by direct substitution that $\sigma_{\delta}^2(\bar{\pi}||\bar{\pi}_{imp}) = 0$ where $\bar{\pi}_{imp}$ is the optimal importance sampling distribution for $\bar{\pi}$ with $\bar{\pi}_{imp}(y|x) \propto \delta(x,y)\bar{\pi}(y|x)$. Nevertheless, informally, the divergence from a logging policy to the target policy is small when the logging policy assigns similar propensities to (x,y) pairs as the importance sampling distribution for the target policy. Conversely, if the logging policy deviates significantly from the importance sampling distribution, then the divergence is large. Based on this notion of divergence, we propose the following weighted estimator:

Definition 6.2 (Weighted IPS Estimator). Assume $\sigma_{\delta}^2(\bar{\pi}||\pi_i) > 0$ for all $1 \leq i \leq m$.

$$
\hat{U}_{weight}(\bar{\pi}) = \sum_{i=1}^{m}\lambda_i^*\sum_{j=1}^{n_i}\frac{\delta_j^i\bar{\pi}(y_j^i|x_j^i)}{p_j^i}
$$

where the weights λ_i^* are set to

$$
\lambda_i^* = \frac{1}{\sigma_{\delta}^2(\bar{\pi}||\pi_i)\sum_{j=1}^{m}\frac{n_j}{\sigma_{\delta}^2(\bar{\pi}||\pi_j)}}. \tag{4}
$$

Note that the assumption $\sigma_{\delta}^2(\bar{\pi}||\pi_i) > 0$ is easily satisfied as long as the logging policy is not exactly equal to the optimal importance sampling distribution of the target policy $\bar{\pi}$. This is very unlikely given that the utility of the new policy is unknown to us in the first place.

We will show that the Weighted IPS Estimator is optimal in the sense that any other convex combination by λ_i that ensures unbiasedness does not give a smaller variance estimator. First, we have a simple condition for unbiasedness:

Proposition 6.3 (Bias of Weighted IPS Estimator). *Assume each logging policy π_i has support for target policy $\bar{\pi}$. Consider the estimator*

$$\hat{U}_{\lambda}(\bar{\pi}) = \sum_{i=1}^{m} \lambda_i \sum_{j=1}^{n_i} \frac{\delta_j^i \bar{\pi}(y_j^i|x_j^i)}{p_j^i}$$

such that $\lambda_i \geq 0$ and $\sum_{i=1}^{m} \lambda_i n_i = 1$. For $\mathcal{D}$ consisting of i.i.d. draws from $\Pr(\mathcal{X})$ and logging policies $\pi_i(\mathcal{Y}|x)$, the above estimator is unbiased:

$$\mathbb{E}_{\mathcal{D}}[\hat{U}_{\lambda}(\bar{\pi})] = U(\bar{\pi}).$$

In particular, $\hat{U}_{weight}(\bar{\pi})$ is unbiased.

Proof. Following the proof of Proposition 4.3,

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[\hat{U}_{\lambda}(\bar{\pi})] &= \sum_{i=1}^{m} \lambda_i \sum_{j=1}^{n_i} \mathbb{E}_{x \sim \Pr(\mathcal{X}), y \sim \pi_i(\mathcal{Y}|x)} \left[\frac{\delta(x,y)\bar{\pi}(y|x)}{\pi_i(y|x)} \right] \\ &= U(\bar{\pi}) \sum_{i=1}^{m} \lambda_i n_i = U(\bar{\pi}).\end{aligned}$$

Moreover, $\sum_{i=1}^{m} \lambda_i^* n_i = 1$, which implies $\hat{U}_{weight}(\bar{\pi})$ is unbiased. □

Notice that making the weights equal reduces $\hat{U}_{\lambda}(\bar{\pi})$ to $\hat{U}_{naive}(\bar{\pi})$. Furthermore, dropping samples from logging policy π_i is equivalent to setting $\lambda_i = 0$.

To prove variance optimality, note that the variance of the Weighted IPS Estimator for a given set of weights $\lambda_1, ..., \lambda_m$ can be written in terms of the divergences.

$$\mathrm{Var}_{\mathcal{D}}[\hat{U}_{\lambda}(\bar{\pi})] = \sum_{i=1}^{m} {\lambda_i}^2 n_i \sigma_{\delta}^2(\bar{\pi}||\pi_i). \tag{5}$$

We now prove the following theorem:

Theorem 6.4. *Assume each logging policy π_i has support for target policy $\bar{\pi}$, and $\sigma_{\delta}^2(\bar{\pi}||\pi_i) > 0$. Then, for any estimator of the form $\hat{U}_{\lambda}(\bar{\pi})$ as defined in Proposition 6.3*

$$\mathrm{Var}_{\mathcal{D}}[\hat{U}_{weight}(\bar{\pi})] = \frac{1}{\sum_{i=1}^{m} \frac{n_i}{\sigma_{\delta}^2(\bar{\pi}||\pi_i)}} \leq \mathrm{Var}_{\mathcal{D}}[\hat{U}_{\lambda}(\bar{\pi})].$$

Proof. The expression for the variance of $\hat{U}_{weight}(\bar{\pi})$ can be verified to be as stated by directly substituting λ_i^* (4) into the variance expression in Equation (5). Next, by the Cauchy-Schwarz inequality,

$$\left(\sum_{i=1}^{m} {\lambda_i}^2 n_i \sigma_{\delta}^2(\bar{\pi}||\pi_i)\right)\left(\sum_{i=1}^{m} \frac{n_i}{\sigma_{\delta}^2(\bar{\pi}||\pi_i)}\right) \geq \left(\sum_{i=1}^{m} \lambda_i n_i\right)^2 = 1$$

$$\Rightarrow \mathrm{Var}_{\mathcal{D}}[\hat{U}_{\lambda}(\bar{\pi})] \geq \mathrm{Var}_{\mathcal{D}}[\hat{U}_{weight}(\bar{\pi})]$$

□

Returning to the toy example in Table 1, the divergence values are $\sigma_{\delta}^2(\bar{\pi}||\pi_1) \approx 252.81$ and $\sigma_{\delta}^2(\bar{\pi}||\pi_2) \approx 4.27$. This leads to weights $\lambda_1^* \approx 0.02$ and $\lambda_2^* \approx 0.98$, resulting in $\mathrm{Var}_{\mathcal{D}}[\hat{U}_{weight}(\bar{\pi})] \approx 4.19 < 4.27$ on $\mathcal{D} = \mathcal{D}^1 \bigcup \mathcal{D}^2$. Thus, the weighted IPS estimator does better than the naive IPS estimator (including the case when $\mathcal{D}^1$ is dropped) by optimally weighting all the available data.

Note that computing the optimal weights λ_i exactly requires access to the utility function δ everywhere in order to compute the divergences $\sigma_{\delta}^2(\bar{\pi}||\pi_i)$. However, in practice, δ is only known at the collected data samples, and the weights must be estimated. In Section 7.6 we discuss a simple strategy for doing so, along with an empirical analysis of the procedure.

6.1 Quantifying the Variance Reduction

The extent of variance reduction provided by the Weighted IPS Estimator over the Naive IPS Estimator depends only on the relative proportions of divergences and the log data sizes of each logging policy. The following proposition quantifies the variance reduction.

Proposition 6.5. *Let $v_i = \frac{\sigma_{\delta}^2(\bar{\pi}||\pi_i)}{\sigma_{\delta}^2(\bar{\pi}||\pi_m)}$ be the ratio of divergences and $r_i = \frac{n_i}{n_m}$ be the ratio of sample sizes of policy i and policy m. Then the reduction denoted as γ is*

$$\gamma \equiv \frac{\mathrm{Var}_{\mathcal{D}}[\hat{U}_{weight}(\bar{\pi})]}{\mathrm{Var}_{\mathcal{D}}[\hat{U}_{naive}(\bar{\pi})]} = \frac{(\sum_{i=1}^{m} r_i)^2}{(\sum_{i=1}^{m} r_i v_i)(\sum_{i=1}^{m} \frac{r_i}{v_i})} \leq 1.$$

Proof. Substituting the expressions for the two variances, we get that

$$\frac{\mathrm{Var}_{\mathcal{D}}[\hat{U}_{weight}(\bar{\pi})]}{\mathrm{Var}_{\mathcal{D}}[\hat{U}_{naive}(\bar{\pi})]} = \frac{(\sum_{i=1}^{m} n_i)^2}{(\sum_{i=1}^{m} n_i \sigma_{\delta}^2(\bar{\pi}||\pi_i))(\sum_{i=1}^{m} \frac{n_i}{\sigma_{\delta}^2(\bar{\pi}||\pi_i)})}$$

So, normalizing by $\sigma_{\delta}^2(\bar{\pi}||\pi_n)$ and n_n, gives the desired expression. Applying the Cauchy-Schwarz inequality gives the upper bound. □

For the case of just two logging policies, $n = 2$, it is particularly easy to compute the maximum improvement in variance of the Weighted IPS Estimator over the Naive estimator. The reduction γ is $\gamma = \frac{(r_1+1)^2 v_1}{(r_1 v_1 + 1)(r_1 + v_1)}$, which ranges between 0 and 1 depending on r_1 and v_1. The benefit of the weighted estimator over the naive estimator is greatest when the logging policies differ substantially, and there are equal amounts of log data from the two logging policies. Intuitively, this is because the weighted estimator mitigates the defect in the naive estimator due to which abundant high variance samples drown out the signal from the equally abundant low variance samples. On the other hand, the scope for improvement is less when the logging policies are similar or when there are disproportionately many samples from one logging policy.

7 EMPIRICAL ANALYSIS

In this section, we empirically examine the properties of the proposed estimators. To do this, we create a controlled setup in which we have logging policies of different utilities, and try to estimate the utility of a fixed new policy. We illustrate key properties of our estimators in the concrete setting of CRF policies for multi-label classification, although the estimators themselves are applicable to arbitrary stochastic policies and structured output spaces.

7.1 Setup

We choose multi-label classification for our experiments because of the availability of a rich feature space $\mathcal{X}$ and an easily scalable label space $\mathcal{Y}$. Three multi-label datasets from the LibSVM repository

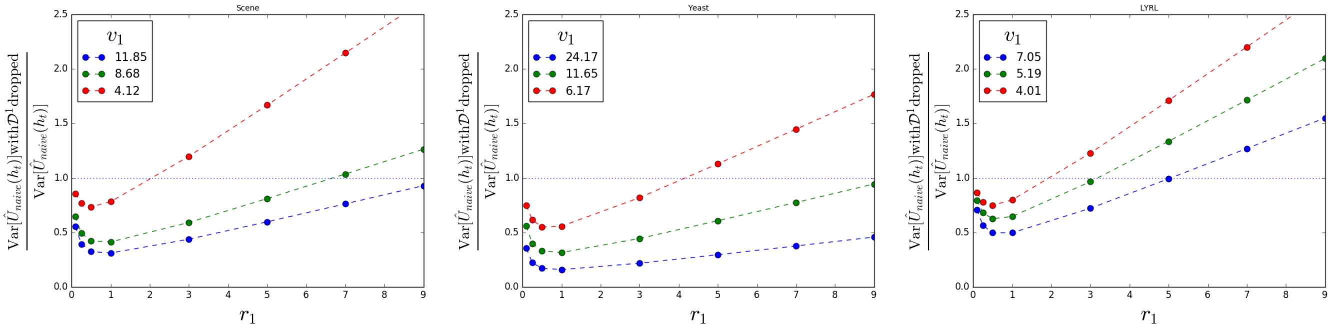

Figure 1: Variance of the Naive IPS Estimator using only π_2 relative to the variance of the Naive IPS Estimator using data from both π_1 and π_2 for different π_1 as the relative sample size changes. Dropping data can lower the variance of Naive IPS Estimator in many cases.

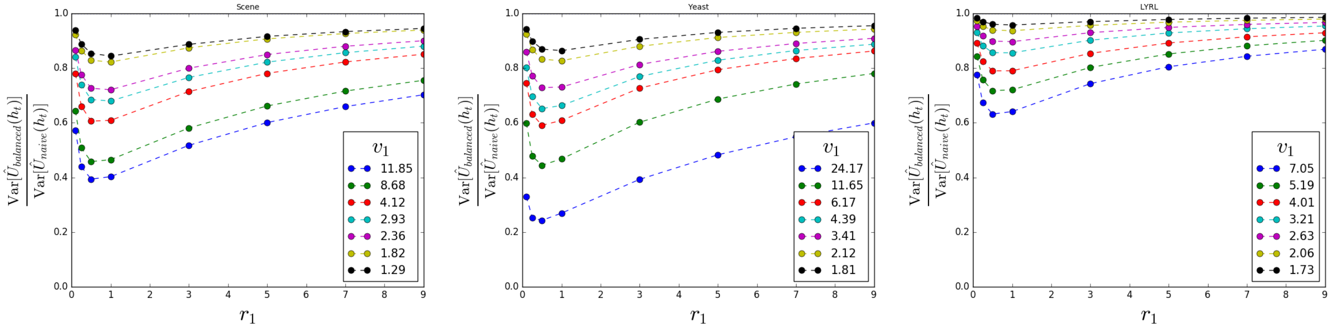

Figure 2: Variance of the Balanced IPS Estimator relative to the variance of the Naive IPS Estimator for different π_1 as the relative sample size changes. The Balanced IPS Estimator can have substantially smaller variance than the Naive IPS Estimator.

Name	# features	# labels	n_{train}	n_{test}
Scene	294	6	1211	1196
Yeast	103	14	1500	917
LYRL	47236	4	23149	781265

Table 2: Corpus statistics for different multi-label datasets from the LibSVM repository. LYRL was post-processed so that only top level categories were treated as labels

with varying feature dimensionalities, number of class labels, and number of training samples available are used. The corpus statistics are as summarized in Table 2.

Since these datasets involve multi-label classification, the output space is $\mathcal{Y} = \{0,1\}^q$, i.e., the set of all possible labels one can generate given a set of q labels. The input distribution $\Pr(\mathcal{X})$ is the empirical distribution of inputs as represented in the test set. The utility function $\delta(x, y)$ is simply the number of correctly assigned labels in y with respect to the given ground truth label y^*.

To obtain policies with different utilities in a systematic manner, we train conditional random fields (CRFs) on incrementally varying fractions of the labeled training set. CRFs are convenient since they provide explicit probability distributions over possible predictions conditioned on an input. However, nothing in the following analysis is specific to using CRFs as the stochastic logging policies, and note that the target policy need not be stochastic at all.

For simplicity and ease of interpretability, we use two logging policies in the following experiments. To generate these logging policies, we vary the training fraction for the first logging policy π_1 over 0.02, 0.05, 0.08, 0.11, 0.14, 0.17, 0.20, keeping the training fractions for the second logging policy π_2 fixed at 0.30. Similarly, we generate a CRF classifier representing the target policy $\bar{\pi}$ by training on 0.35 fraction of the data. The effect is that we now get three policies where the second logging policy is similar to the target while the similarity of the first logging policy varies over a wide range. This results in a wide range of relative divergences

$$v_1 = \frac{\sigma_\delta^2(\bar{\pi}||\pi_1)}{\sigma_\delta^2(\bar{\pi}||\pi_2)}$$

for the first logging policy on which the relative performance of the estimators depends.

We compare pairs of estimators based on their relative variance since all the estimators being considered are unbiased (so, relative variance 1 signifies the estimators being compared have the same variance). Since the variance of the different estimators scales inversely proportional to the total number of samples, the ratio of their variances depends only on the relative size of the two data logs

$$r_1 = \frac{n_1}{n_2},$$

but not on their absolute size. We therefore report results in terms of relative size where we vary $r_1 \in \{0.1, 0.25, 0.5, 1, 3, 5, 7, 9\}$ to explore a large range of data imbalances.

For a fixed set of CRFs as logging and target policies, and the relative size of the data logs, the ratio of the variances of the different estimators can be computed exactly since the CRFs provide explicit distributions over $\mathcal{Y}$, and $\mathcal{X}$ is based on the test set. We therefore report exact variances in the following. In addition to the exactly computed variances, we also did some bandit feedback simulations to verify the experiment setup. We employed the Supervised $\mapsto$ Bandit conversion method [1]. In this method, we iterate over the test features x, sample some prediction y from the logging policy $\pi_i(\mathcal{Y}|x)$ and record the corresponding loss and propensity to generate the logged data-sets $\mathcal{D}^i$. For various settings of logging policies and amounts of data, we sampled bandit data and obtained estimator values over hundreds of iterations. We then computed the empirical mean and variance of the different estimates to make sure that the estimators were indeed unbiased and closely matched the theoretical variances reported above.

7.2 Can dropping data lower the variance of $\hat{U}_{naive}(\bar{\pi})$?

While we saw that dropping data improved the variance of the Naive IPS Estimator in the toy example, we first verify that this issue also surfaces outside of carefully constructed toy examples. To this effect, Figure 1 plots the variance of the Naive IPS Estimator $\hat{U}_{naive}(\bar{\pi})$ that uses data only from π_2 relative to the variance of $\hat{U}_{naive}(\bar{\pi})$ when using data from both π_1 and π_2. The x-axis varies the relative amount of data coming from π_1 and π_2. Each solid circle on the plot corresponds to a training fraction choice for π_1 and a log-data-size ratio r_1. A lot-data-size ratio of 0 means that no data from π_1 is used, i.e., all data from π_1 is dropped. The relative divergence v_1 is higher when π_1 is trained on a lower fraction of training data since in that case π_1 differs more from π_2. A solid circle below the baseline at 1 indicates that dropping data improves the variance in that case.

Overall, the experiments confirm that the Naive IPS Estimator shows substantial inefficiency. We observe that for high v_1 and small r_1, dropping data from π_1 can reduce the variance substantially for a wide range of realistic CRF policies. As v_1 decreases and r_1 increases, dropping data becomes less beneficial, ultimately becoming worse than the using all the data. This concurs with the intuition that dropping a relatively small number of high variance data samples can help utilize the low variance data samples.

7.3 How does $\hat{U}_{bal}(\bar{\pi})$ compare with $\hat{U}_{naive}(\bar{\pi})$?

We proved that the Balanced IPS Estimator has smaller (or equal) variance than the Naive IPS Estimator. The experiments reported in Figure 2 show the magnitude of variance reduction for $\hat{U}_{bal}(\bar{\pi})$. In particular, Figure 2 reports the variance of the Balanced IPS Estimator relative to the variance of the Naive IPS Estimator for different logging policies π_1 and different data set imbalances. In all cases, $\hat{U}_{bal}(\bar{\pi})$ performs at least as well as $\hat{U}_{naive}(\bar{\pi})$ and the variance reduction increases when the two policies differ more (i.e. v_1 is large). The variance reduction due to $\hat{U}_{bal}(\bar{\pi})$ decreases as the relative size of the log data from π_1 increases.

7.4 How does $\hat{U}_{weight}(\bar{\pi})$ compare with $\hat{U}_{naive}(\bar{\pi})$?

We know that the Weighted IPS Estimator always has lower variance (or equal) than the Naive IPS Estimator. The results in Figure 3 show the magnitude of the relative variance improvement for the Weighted IPS Estimator. As in the case of the Balanced IPS Estimator, $\hat{U}_{weight}(\bar{\pi})$ performs better than $\hat{U}_{naive}(\bar{\pi})$ especially when the two logging policies differ substantially. This confirms the theoretical characterization of $\hat{U}_{weight}(\bar{\pi})$ from Section 6.1, where we computed the variance reduction given r_1 and v_1. The empirical findings are as expected by the theory and show a substantial improvement in this realistic setting. However, note that these experiments do not yet address the question of how to estimate the weights in practice, which we come back to in Section 7.6.

7.5 How does $\hat{U}_{weight}(\bar{\pi})$ compare with $\hat{U}_{bal}(\bar{\pi})$?

We did not find theoretical arguments whether $\hat{U}_{weight}(\bar{\pi})$ is uniformly better than $\hat{U}_{bal}(\bar{\pi})$ or vice versa. The empirical results in Figure 4 confirm that either estimator can be preferable in some situations. Specifically, $\hat{U}_{weight}(\bar{\pi})$ performs better when the difference between the two logging policies is large, whereas $\hat{U}_{bal}(\bar{\pi})$ performs better when they are closer. This is an interesting phenomenon that merits future investigation. In particular, one might be able to combine the strengths of $\hat{U}_{weight}(\bar{\pi})$ and $\hat{U}_{bal}(\bar{\pi})$ to get a weighted form of the $\hat{U}_{bal}(\bar{\pi})$ estimator. Since we know from the toy example that even $\hat{U}_{bal}(\bar{\pi})$ can have lower variance with dropping data, it is plausible that it could improve if the samples were weighted non-uniformly.

7.6 How can we estimate the weights for $\hat{U}_{weight}(\bar{\pi})$?

We derived the optimal weights λ_i^* in terms of $\sigma_\delta^2(\bar{\pi}||\pi_i)$. Computing the divergence exactly requires access to the utility function $\delta(x, y)$ on the entire domain $\mathcal{X} \times \mathcal{Y}$. However, $\delta(x, y)$ is known only at the samples collected as bandit feedback. We propose the following strategy to estimate the weights in this situation.

Each divergence can be estimated by using the empirical variance of the importance-weighted utility values available in the log data $\mathcal{D}^i$.

$$\hat{\sigma}_\delta^2(\bar{\pi}||\pi_i) = \hat{\mathrm{Var}}_{\mathcal{D}^i}\left[\frac{\delta_j^i \cdot \bar{\pi}(y_j^i|x_j^i)}{p_j^i}\right]$$

Under mild conditions, this provides a consistent estimate since $x_j^i \sim \Pr(\mathcal{X})$ and $y_j^i \sim \pi_i(\mathcal{Y}|x_j^i)$. The weights λ_i are then obtained using the estimated divergences.

We tested this method by generating bandit data using the Supervised $\mapsto$ Bandit conversion method described in Section 7.1 for each logging policy, and then computing the weights as described above. Figure 5 compares the variance of the weighted estimator with the estimated weights against the variance with the optimal weights. The x-axis varies the size of the log data for both logging

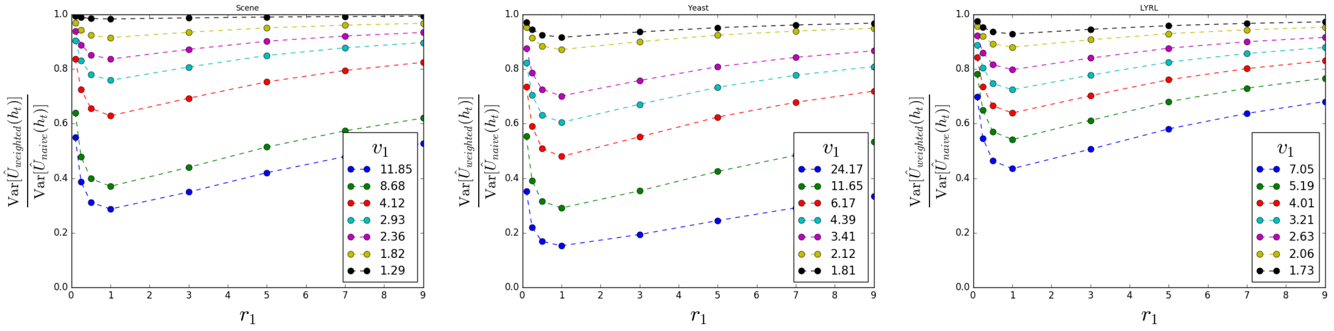

Figure 3: Variance of the Weighted IPS Estimator relative to the variance of the Naive IPS Estimator for different π_1 as the relative sample size changes. The Weighted IPS Estimator can have substantially smaller variance than the Naive IPS Estimator.

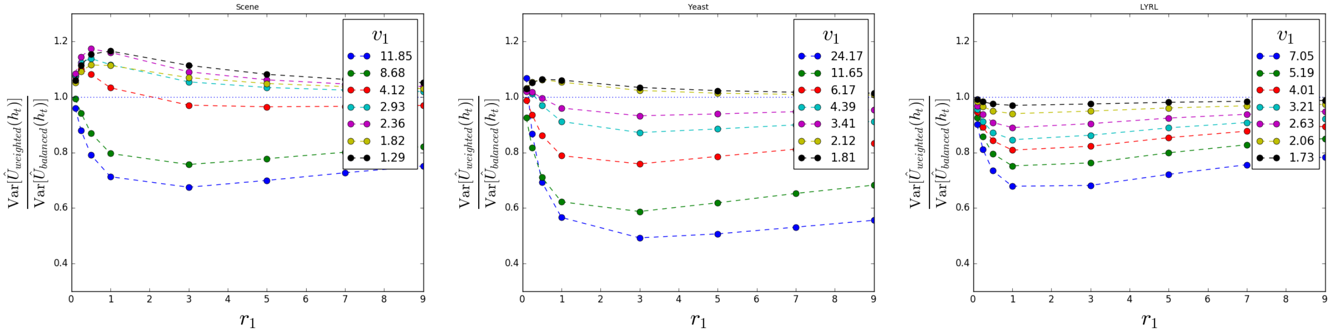

Figure 4: Variance of the Weighted IPS Estimator relative to the variance of the Balanced IPS Estimator for different π_1 as the relative sample size changes. The Weighted IPS Estimator does better than the Balanced IPS Estimator when the two logging policies differ significantly. However, the Balanced IPS Estimator performs better when the two policies are similar.

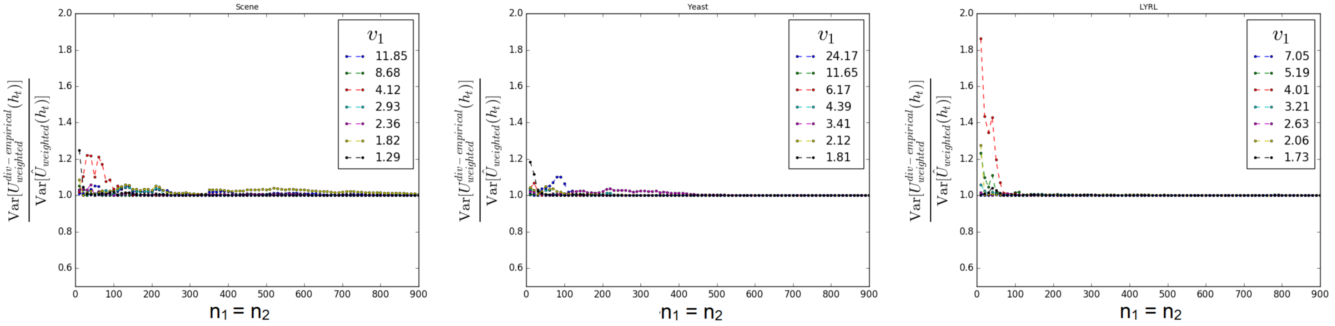

Figure 5: Variance with weights estimated from empirical divergences relative to optimal weights for the Weighted IPS Estimator. The estimation works very well when there is sufficient amount of log data. We chose $m_1 = m_2$, i.e. $r_1 = 1$ for convenience. Similar trends were observed for other values of r_1.

policies π_1 and π_2 which are kept equal (i.e. $n_1 = n_2$) for simplicity. As shown, the variance of the estimator with the estimated weights converges to that of the optimal weighted estimator within a few hundred samples for all choices of logging policies and across the three data-sets. Similar trends were observed for other values of relative log data size r_1 as well.

Note that in this method we take the empirical variance of the importance-weighted utility values over each log $\mathcal{D}^i$ individually to get reliable unbiased estimates of the true divergences. In contrast, the Naive IPS Estimator takes the empirical mean of the same values over the combined data $\mathcal{D}$. Therefore, the former estimation does not suffer from the suboptimality in variance that occurs due to naively combining data from different logging policies.

Therefore, we conclude that the above method of estimating the weights performs quite well and seems well suited for practical applications.

8 CONCLUSION

We investigated the problem of estimating the performance of a new policy using data from multiple logging policies in a contextual bandit setting. This problem is highly relevant for practical applications since it reflects how logged contextual bandit feedback is available in online systems that are frequently updated (e.g. search engines, ad placement systems, product recommenders). We proposed two estimators for this problem which are provably unbiased and have lower variance than the Naive IPS Estimator. We empirically demonstrated that both can substantially reduce variance across a range of evaluation scenarios.

The findings raise interesting questions for future work. First, it is plausible that similar estimators and advantages also exist for other partial-information data settings [9] beyond contextual bandit feedback. Second, while this paper only considered the problem of evaluating a fixed new policy $\bar{\pi}$, it would be interesting to use the new estimators also for learning. In particular, they could be used to replace the Naive IPS Estimator when learning from bandit feedback via Counterfactual Risk Minimization [21].

ACKNOWLEDGMENTS

This work was supported in by under NSF awards IIS-1615706 and IIS-1513692, and through a gift from Bloomberg. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1650441. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert E. Schapire. 2014. Taming the monster: A fast and simple algorithm for contextual bandits. In *In Proceedings of the 31st International Conference on Machine Learning (ICML-14.* 1638–1646.

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

[3] Ben Carterette, Virgil Pavlu, Evangelos Kanoulas, Javed A. Aslam, and James Allan. 2009. If I Had a Million Queries. In *ECIR*. 288–300.

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

[5] Víctor Elvira, Luca Martino, David Luengo, and Mónica F Bugallo. 2015. Efficient multiple importance sampling estimators. *IEEE Signal Processing Letters* 22, 10 (2015), 1757–1761.

[6] Víctor Elvira, Luca Martino, David Luengo, and Mónica F Bugallo. 2015. Generalized multiple importance sampling. *arXiv preprint arXiv:1511.03095* (2015).

[7] Hera Y. He and Art B. Owen. 2014. Optimal mixture weights in multiple importance sampling. (2014). arXiv:arXiv:1411.3954

[8] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. 2013. Reusing historical interaction data for faster online learning to rank for IR. In *Proceedings of the sixth ACM international conference on Web search and data mining*. 183–192.

[9] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)*. ACM, New York, NY, USA, 781–789. DOI: http://dx.doi.org/10.1145/3018661.3018699

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

[11] 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*. ACM, 929–934.

[12] 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*. ACM, 661–670.

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

[14] Lihong Li, Jin Young Kim, and Imed Zitouni. 2015. Toward predicting the outcome of an A/B experiment for search relevance. In *WSDM*. 37–46.

[15] Art Owen and Yi Zhou. 2000. Safe and effective importance sampling. *J. Amer. Statist. Assoc.* 95, 449 (2000), 135–143.

[16] Art B. Owen. 2013. *Monte Carlo theory, methods and examples.*

[17] Doina Precup. 2000. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series* (2000), 80.

[18] Tobias Schnabel, Adith Swaminathan, Peter I. Frazier, and Thorsten Joachims. 2016. Unbiased Comparative Evaluation of Ranking Functions. In *ICTIR*. 109–118.

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

[20] Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*. Vol. 1. MIT press Cambridge.

[21] Adith Swaminathan and Thorsten Joachims. 2015. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of the 32nd International Conference on Machine Learning*. 814–823.

[22] Eric Veach and Leonidas J Guibas. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *SIGGRAPH*. 419–428.

[23] Emine Yilmaz, Evangelos Kanoulas, and Javed A. Aslam. 2008. A Simple and Efficient Sampling Method for Estimating AP and NDCG. In *SIGIR*. 603–610.