Off-Policy Evaluation with Deficient Support Using Side Information

Nicolò Felicioni
Politecnico di Milano
nicolo.felicioni@polimi.it

Maurizio Ferrari Dacrema
Politecnico di Milano
maurizio.ferrari@polimi.it

Marcello Restelli
Politecnico di Milano
marcello.restelli@polimi.it

Paolo Cremonesi
Politecnico di Milano
paolo.cremonesi@polimi.it

Abstract

The Off-Policy Evaluation (OPE) problem consists in evaluating the performance of new policies from the data collected by another one. OPE is crucial when evaluating a new policy online is too expensive or risky. Many of the state-of-the-art OPE estimators are based on the Inverse Propensity Scoring (IPS) technique, which provides an unbiased estimator when the full support assumption holds, i.e., when the logging policy assigns a non-zero probability to each action. However, there are several scenarios where this assumption does not hold in practice, i.e., there is deficient support, and the IPS estimator is biased in the general case. In this paper, we consider two alternative estimators for the deficient support OPE problem. We first show how to adapt an estimator that was originally proposed for a different domain to the deficient support setting. Then, we propose another estimator, which is a novel contribution of this paper. These estimators exploit additional information about the actions, which we call side information, in order to make reliable estimates on the unsupported actions. Under alternative assumptions that do not require full support, we show that the considered estimators are unbiased. We also provide a theoretical analysis of the concentration when relaxing all the assumptions. Finally, we provide an experimental evaluation showing how the considered estimators are better suited for the deficient support setting compared to the baselines.

1 Introduction

Many real-world decision-making problems can be viewed through the lens of the contextual bandit framework. Some prominent examples are medical treatments [19], recommendation systems [15, 18, 31, 48], search engines [32], ad-placement systems [9], and many others. In any of these problems, we have a decision-maker who repeatedly observes a context (e.g., a profile of a patient), samples an action according to its policy (e.g., provides a medical treatment), and collects a reward (e.g., +1 if the patient survives). Also, in many of those applications, we have access to additional information about the actions, which we call *side information*. For example, in movie recommendation, we may know the director and the actors of each movie [12]; in a clinical trial, we may have access to a set of characteristics of each drug [10].

While deploying and evaluating a new policy on a real system may be prohibitively expensive (if not unfeasible), logged data of contextual bandit problems is relatively cheap to obtain. Hence, it is desirable to use them to evaluate new policies, without the need to collect new data. This problem is called *Off-Policy Evaluation* (OPE). In this case, we have data collected by a given logging policy, and

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

we want to evaluate a different policy, called evaluation policy. One way to think about this problem is that we are trying to answer the following counterfactual question: "*What would have happened if the evaluation policy was deployed instead of the logging one?*". One of the most widely employed estimators for OPE is the *Inverse Propensity Score* (IPS), along with its variants [43, 48, 56, 61]. The IPS technique tries to de-bias the collected rewards by accounting for the propensity of the logging policy. One of the reasons for the widespread adoption of the IPS estimator is that it is unbiased under the *full support* assumption. The full support assumption states that no action that is possible under the evaluation policy has zero probability under the logging policy. Unfortunately, we argue that this assumption is unrealistic in many real-world systems. A practical example is a recommendation system that adopts a user-based pre-processing of the items to recommend, discarding some items for a given user to improve scalability. In this way, some actions (i.e., items) will never be proposed to a given context (i.e., a user), violating the full support assumption. Another example is the *new action* problem, namely, when the action space is expanded after the logging phase. This problem is typical of many real-world use cases, such as recommendation systems [4, 29, 51, 52], drug interaction evaluation [33], clinical trials [40], etc. In general, whenever the full support assumption is not valid, we say that we have an OPE problem with *deficient support*. If this is the case, the IPS estimator can be highly biased [47]. A possible mitigation for this issue is using a model-based approach, which means training a regression model that aims to approximate the reward function and extrapolate the rewards for the unsupported actions [5, 6, 13, 35, 47]. The drawback of this method is that model misspecification can lead to a high bias [13, 35, 50].

In this paper, we focus on estimators without a regression model for deficient support that exploit side information about the actions. After introducing the necessary background (Section 2), we consider two alternative estimators for Off-Policy Evaluation with deficient support in Section 3. While relaxing the full support assumption, we offer alternative assumptions that hold even with deficient support. In Section 3.1, we present the *PseudoInverse* estimator. This estimator was introduced by Swaminathan et al. [63] in the context of slate recommendation. We show how this estimator can be adapted to the presence of side information, and we show that it is unbiased under two alternative assumptions: *full support on side information* and *linearity*. Full support on side information is a milder assumption than full support: it requires the logging policy to be able to select each feature. Linearity, instead, requires that the rewards are linear combinations of the action features. This assumption is already used in other contexts, such as recommendation systems [27] and online linear bandits [2, 11]. In Section 3.2, we propose a novel estimator, which we call *Similarity* estimator. It is based on the assumption that expected rewards of similar actions are similar. Under this assumption, we prove its unbiasedness, even in the presence of deficient support. In Section 3.3, we relax all the assumptions and derive finite-sample concentration inequalities of the considered estimators for deficient support and the traditional IPS to better understand the theoretical guarantees of each of them. In Section 4, we provide an experimental evaluation, showing that the considered estimators consistently outperform traditional approaches (such as IPS and regression-based estimators) on a real-world dataset with deficient support from the recommendation systems domain. Finally, in Section 5 we review related work, and in Section 6 we draw conclusions.

Societal Impacts Improving OPE may be beneficial to society as it allows decision-makers to assess potentially dangerous policies without testing them on a real system. This is even more relevant in the presence of deficient support, when there are context-action pairs that are never observed in the data.

2 Background and Preliminaries

First of all, we introduce the *Off-Policy Evaluation* (OPE) problem in the contextual bandit setting with side information on the actions, and then, we describe the deficient support problem.

Off-Policy Evaluation Let x be a context sampled from a prior (unknown) context distribution p over the context space X, i.e., $x \sim p(\cdot)$. An action a is sampled from a probability π (called *policy*) over the action space A, conditioned on the observed context x, i.e., $a \sim \pi(\cdot|x)$. After this choice, a reward r is observed from the reward distribution conditioned on the observed context and chosen action, i.e., $r \sim p(\cdot|x, a)$. Without loss of generality, we consider as reward space $[0, 1]$. To evaluate a policy, we use the *policy value* $R(\pi) := \mathbb{E}_{x\sim p(\cdot)} \mathbb{E}_{a\sim\pi(\cdot|x)} \mathbb{E}_{r\sim p(\cdot|x,a)}[r]$. It is useful to define the expected reward given a context and an action: $\delta(x, a) := \mathbb{E}_{r\sim p(\cdot|x,a)}[r]$. With this function, we can simplify the policy value formulation: $R(\pi) = \mathbb{E}_{x \sim p(\cdot)} \mathbb{E}_{a \sim \pi(\cdot|x)}[\delta(x, a)]$. In this paper, we focus on the contextual bandit problem with side information about the actions. Therefore, we have access to action features, which are represented as vectors: $\mathrm{f}(a) \in \mathbb{R}^F$ for any $a \in A$. We focus on the *Off-Policy Evaluation* (OPE) problem. We call the logging policy π_0. Therefore, the collected dataset will be in the following form: $\mathcal{D} := \{x_i, a_i, r_i, \pi_0(a_i|x_i)\}_{i=1}^n$, where, for each i, $x_i \sim p(\cdot)$, $a_i \sim \pi_0(\cdot|x_i)$, and $r_i \sim p(\cdot|x_i, a_i)$. Given this dataset, we would like to evaluate another policy by estimating its value function. The new policy is called the *evaluation policy* π_e. Thus, we want to find a plausible estimator $\hat{R}(\pi_e)$ using the given dataset ($\hat{R}(\pi_e) = \hat{R}(\pi_e; \mathcal{D})$) such that $\hat{R}(\pi_e) \approx R(\pi_e)$. One of the most used estimators for OPE is the *Inverse Propensity Score* (IPS), along with its variants. The IPS estimator is defined as $\hat{R}_{\text{IPS}}(\pi_e) = \frac{1}{n}\sum_{i=1}^n \frac{\pi_e(a_i|x_i)}{\pi_0(a_i|x_i)} r_i$. A fundamental property of such estimator is that it is an *unbiased* estimator of the evaluation policy value if the *full support assumption* holds [31]. Let us define $supp(f) := \{z \mid f(z) > 0\}$, $A_0(x) := supp(\pi_0(\cdot|x))$, and $A_e(x) := supp(\pi_e(\cdot|x))$, for any x. The full support assumption is stated in the following.

Assumption 1 (Full Support). *The off-policy evaluation problem satisfies the full support assumption if $A_e(x) \subseteq A_0(x)$ with probability one for $x \sim p(\cdot)$.*

Deficient Support Whenever the full support assumption is not valid, we say that we have an OPE problem with *deficient support* [47]. If this is the case, the IPS estimator is no longer unbiased. Let us define the set of unsupported actions for context x under π_0 as $\mathrm{Un}(x, \pi_e, \pi_0) := A_e(x) \setminus A_0(x)$. The bias of the IPS depends on this set, as illustrated by the following proposition.

Proposition 1 ([47], Proposition 1). *In an off-policy evaluation problem, the bias of $\hat{R}_{IPS}(\pi_e)$ is equal to the negative expected reward on the unsupported action set:*

$$bias(\hat{R}_{IPS}(\pi_e)) = \mathop{\mathbb{E}}_{x \sim p(\cdot)}\left[-\sum_{a \in Un(x, \pi_e, \pi_0)} \pi_e(a|x)\delta(x, a)\right]$$

This finding is intuitive: if the unsupported set is empty (i.e., the full support assumption is valid), the IPS estimator has no bias. Otherwise, if there are unsupported actions, the IPS estimator has a bias caused by the blind areas in the dataset, where there are actions for which we never observe the reward.

3 Estimators for Off-Policy Evaluation with Side Information

In this section, we first introduce two estimators for a better off-policy evaluation when we have deficient support and side information about the actions, and then we provide finite-sample error bounds. The proofs of the mathematical statements can be found in Appendix A.

3.1 PseudoInverse Estimator

The *PseudoInverse* (PI) estimator was initially proposed by Swaminathan et al. [63] for the problem of off-policy evaluation for *slate* recommendation [23]. In the following, we will show that we can use an analogous estimator for the OPE problem with side information.

We begin by describing two assumptions, which are relatively mild conditions and can also hold in the presence of deficient support.

Assumption 2 (Full Support on Side Information). *The off-policy evaluation problem satisfies the full support on side information if, whenever $\pi_e(a|x) > 0$, this implies that, for every feature of a (i.e., $\forall j \in \{j|\mathrm{f}(a)_j > 0\}$) there exists an action $a' \in A_0(x)$ that has the same feature (i.e., $\mathrm{f}(a')_j > 0$) with probability one over $x \sim p(\cdot)$.*

We notice that this assumption can also be satisfied in the presence of deficient support, as long as the logging policy can propose each feature. Now we make an assumption on the reward structure.

Assumption 3 (Linearity). *For each context $x \in X$ there exists an (unknown) intrinsic reward vector $\phi_x \in \mathbb{R}^F$ such that $\delta(x, a) = \mathrm{f}(a)^T \phi_x$.*

Intuitively, this assumption says that each action feature contributes linearly to the final reward. This assumption found applications in different fields: in recommendation systems, it constitutes the basis of the Matrix Factorization algorithm [27], where we have fixed the latent item factors to be the feature vectors; in bandits literature, it corresponds to the linear bandit formulation [2, 11].

Hence, we can view this problem as a regression problem, where we interpret the ϕ_x vector as the weight vector w to learn and we want to minimize the following error (for each x): $\mathbb{E}_{a \sim \pi_0(\cdot|x)} \mathbb{E}_{r \sim p(\cdot|x,a)}[(\mathrm{f}(a)^T w - r)^2]$. The PseudoInverse estimator derives from the minimization of this error, and it is unbiased under the two previously described assumptions.

Theorem 1. *Consider the off-policy evaluation problem where Assumptions 2 and 3 hold. Then, we can define an unbiased estimator $\hat{R}_{PI}(\pi_e)$ of the expected reward of the evaluation policy as:*

$$\hat{R}_{PI}(\pi_e) := \frac{1}{n} \sum_{i=1}^{n} r_i \cdot \mathbf{q}_{\pi_e, x_i}^T \mathbf{\Gamma}_{\pi_0, x_i}^{\dagger} \mathrm{f}(a_i),$$

where $\mathbf{\Gamma}_{\pi_0,x} := \mathbb{E}_{a \sim \pi_0(\cdot|x)}[\mathrm{f}(a)\mathrm{f}(a)^T] \in \mathbb{R}^{F \times F}$, $\mathbf{q}_{\pi_e,x} := \mathbb{E}_{a \sim \pi_e(\cdot|x)}[\mathrm{f}(a)] \in \mathbb{R}^F$, and $M^\dagger$ is the Moore-Penrose pseudoinverse of a generic matrix M.

For the details of the derivation, we refer to [63]. Inspired by [61], we can also use a *Self-Normalized* version of the estimator, which is a biased but consistent estimator with a reduced variance:

$$\hat{R}_{SN-PI}(\pi_e) := \frac{1}{\sum_{i=1}^{n} \mathbf{q}_{\pi_e, x_i}^T \mathbf{\Gamma}_{\pi_0, x_i}^{\dagger} \mathrm{f}(a_i)} \sum_{i=1}^{n} r_i \cdot \mathbf{q}_{\pi_e, x_i}^T \mathbf{\Gamma}_{\pi_0, x_i}^{\dagger} \mathrm{f}(a_i).$$

3.2 Similarity Estimator

In some cases, the calculation of the pseudoinverse of a matrix can be computationally expensive. Thus, in the following, we propose an estimator based on computationally cheaper operations, which stems from a different assumption. Such assumption comes from the recommendation systems research literature [46]. Two of the most common approaches in recommendation systems are *collaborative filtering* [45] and *content-based filtering* [36] methods. Both of these methods are based on the assumption that the reward of an item (i.e., an action in the contextual bandit setting) given a user (i.e., a context) can be approximated by a weighted sum of rewards observed from different items given the same user. In our setting, we can define this assumption as follows:

Assumption 4. *The off-policy evaluation problem satisfies the similarity assumption if, given any context $x \in X$, the following condition holds:*

$$\exists w_x : A_e(x) \times A_0(x) \to \mathbb{R} \ \textit{such that} \ \forall a \in A_e(x),\ \delta(x,a) = \sum_{a' \in A_0(x)} w_x(a,a')\delta(x,a')$$
$$\textit{and} \sum_{a' \in A_0(x)} w_x(a,a') = 1.$$

This indicates that we can approximate the reward for any action with a weighted sum of rewards observed from the actions supported by the logging policy. The weighting function should sum up to one over the supported actions, and intuitively it should indicate how much two actions are perceived as similar for a given context.

The aforementioned assumption is also strictly tied with the *Lipschitz assumption*, which is widely exploited in the research field of online bandits with side information [25, 26, 41, 54, 55]. The connection between the two assumptions is exhibited by the Proposition 3, in Appendix A.

By exploiting the similarity assumption, we can get an unbiased estimator of the expected reward of the evaluation policy even with deficient support. We explain the procedure in the following.

Theorem 2. *Consider the off-policy evaluation problem where the similarity assumption holds. Then, we can define an unbiased estimator $\hat{R}_S(\pi_e)$ of the expected reward of the evaluation policy as:*

$$\hat{R}_S(\pi_e) := \frac{1}{n} \sum_{i=1}^{n} \frac{\bar{\pi}(a_i|x_i)}{\pi_0(a_i|x_i)} r_i,$$

where $\bar{\pi}(a_i|x_i) := \mathbb{E}_{a \sim \pi_e(\cdot|x_i)}\left[w_{x_i}(a, a_i)\right]$.

We notice that the meaning of the estimator is somehow intuitive. Each observed reward r_i is divided by the propensity of the logging policy to eliminate the bias introduced by the data collection procedure (as in IPS). At the same time, it is amplified by how much, on average, the evaluation policy proposes an action similar to a_i for the context x_i. Given that its structure is comparable to the one of the IPS, the Self-Normalized version of the Similarity estimator is a consistent estimator under the same assumptions of Theorem 2 [61]:

$$\hat{R}_{SN-S}(\pi_e) := \frac{1}{\sum_{i=1}^{n} \frac{\bar{\pi}(a_i|x_i)}{\pi_0(a_i|x_i)}} \sum_{i=1}^{n} \frac{\bar{\pi}(a_i|x_i)}{\pi_0(a_i|x_i)} r_i.$$

Comparison with Regression-based Estimators While this estimator does not require a regression model, we show that, from a theoretical point of view, the similarity estimator has some analogies with a regression-based estimator. Within this analysis, we will also see the essential differences between the two types of estimators. First, we define the simplest and most used type of regression-based estimator, which we call *Direct Method* (DM) [5]:

$$\hat{R}_{DM}(\pi_e) = \frac{1}{n} \sum_{i=1}^{n} \sum_{a \in A_e(x_i)} \pi_e(a|x_i) \hat{\delta}(x_i, a),$$

where $\hat{\delta}$ is a regression model. The theoretical analysis will focus on the expected values of the estimators. To lighten the notation, we will omit the distributions of the expected value whenever the expectation is with respect to the data distribution. The expected value of DM is:

$$\mathbb{E}[\hat{R}_{DM}(\pi_e)] = \mathop{\mathbb{E}}_{x \sim p(\cdot)} \mathop{\mathbb{E}}_{a \sim \pi_e(\cdot|x)} \left[\hat{\delta}(x, a) \right].$$

Now, let us analyze the expected value of the similarity estimator:

$$\begin{aligned}
\mathbb{E}[\hat{R}_S(\pi_e)] &= \mathop{\mathbb{E}}_{x \sim p(\cdot)} \mathop{\mathbb{E}}_{a' \sim \pi_0(\cdot|x)} \left[\frac{\bar{\pi}(a'|x)}{\pi_0(a'|x)} \delta(x, a') \right] \\
&= \mathop{\mathbb{E}}_{x \sim p(\cdot)} \left[\sum_{a' \in A_0(x)} \bar{\pi}(a'|x) \delta(x, a') \right] \\
&= \mathop{\mathbb{E}}_{x \sim p(\cdot)} \left[\sum_{a' \in A_0(x)} \sum_{a \in A_e(x)} \pi_e(a|x) w_x(a, a') \delta(x, a') \right] \quad \text{(from the definition of } \bar{\pi}) \\
&= \mathop{\mathbb{E}}_{x \sim p(\cdot)} \mathop{\mathbb{E}}_{a \sim \pi_e(\cdot|x)} \left[\sum_{a' \in A_0(x)} w_x(a, a') \delta(x, a') \right].
\end{aligned}$$

Therefore, we notice that the similarity estimator is analogous (in expectation) to a regression-based estimator where the regression model is $\hat{\delta}(x, a) = \sum_{a' \in A_0(x)} w_x(a, a') \delta(x, a')$. What the similarity estimator is doing is trying to overcome support deficiency by exploiting the expected reward on supported actions and the information on the similarity between supported and unsupported actions.

However, it would be erroneous to think that, because of this analogy, it suffices to create a regression-based estimator with the regression model $\hat{\delta}(x, a) = \sum_{a' \in A_0(x)} w_x(a, a') \delta(x, a')$. The problem is that, in general, it is not possible to directly estimate all the possible expected rewards $\delta(x, a')$ for any $a' \in A_0(x)$ from data. Indeed, the fact that $a' \in A_0(x)$ does not imply that the logging policy chose action a', but only that the probability of selecting a' was positive. This means that, in general, we may have never seen a pair x, a' in the dataset, and consequently, we may have never observed any $r \sim p(\cdot|x, a')$. Therefore, the only way to exploit the similarity assumption is via the proposed Similarity estimator.

3.3 Concentration Analysis

Up until now, we have shown that the previous estimators are unbiased under some assumptions. In the following, we will analyze the concentration behavior of the PseudoInverse, Similarity, and traditional

IPS estimators. We will also relax every assumption and investigate the bias-variance trade-off that arises. To do so, we provide finite-sample error bounds. In order to lighten the notation, we will drop the distributions of the expected value whenever the expectation is for $(x, a, r) \sim p(\cdot)\pi_0(\cdot|x)p(\cdot|x, a)$. We will refer to the absolute value of the bias of an estimator as $\epsilon := |\,\mathbb{E}[\hat{R}(\pi_e)] - R(\pi_e)|$. We introduce two different functions to measure the difference between the conditional distributions induced by two policies. The first one is a premetric called *Support Divergence*, introduced by Sachdeva et al. [47]:

$$d^{supp}(\pi_e||\pi_0) := \mathop{\mathbb{E}}_{x\sim p(\cdot)}\left[\sum_{a\in\mathrm{Un}(x,\pi_e,\pi_0)} \pi_e(a|x)\right].$$

The second one is the *Exponentiated Rényi Divergence* [37, 44]:

$$d_2(\pi_e||\pi_0) := \mathop{\mathbb{E}}_{\substack{x\sim p(\cdot)\\ a\sim\pi_0(\cdot|x)}}\left[\left(\frac{\pi_e(a|x)}{\pi_0(a|x)}\right)^2\right].$$

Now, we are ready to introduce finite-sample concentration inequalities.

Proposition 2. *Let us consider a generic off-policy evaluation problem with side information. Let $\hat{R}_{IPS}$ be the IPS estimator, $\hat{R}_S$ the similarity estimator with weighting function w_x, $\hat{R}_{PI}$ the pseudo-inverse estimator. Then, for any $\gamma \in (0, 1)$, the following inequalities hold with probability at least $1 - \gamma$:*

$$\left|\hat{R}_{IPS}(\pi_e) - R(\pi_e)\right| \le d^{supp}(\pi_e||\pi_0) + \sqrt{\frac{d_2(\pi_e||\pi_0)}{n\gamma}},$$

$$\left|\hat{R}_S(\pi_e) - R(\pi_e)\right| \le \epsilon^S + \sqrt{\frac{d_2(\bar{\pi}||\pi_0)}{n\gamma}},$$

$$\left|\hat{R}_{PI}(\pi_e) - R(\pi_e)\right| \le \epsilon^{PI} + \sqrt{\frac{2\sigma^2_{PI}\ln(2/\gamma)}{n}} + \frac{2(\rho_{PI}+1)\ln(2/\gamma)}{3n},$$

where $\bar{\pi}(a|x) := \mathbb{E}_{a'\sim\pi_e(\cdot|x)}[w_x(a', a)]$, $\sigma^2_{PI} := \mathbb{E}_{x\sim p(\cdot)}[\mathbf{q}^T_{\pi_e,x}\mathbf{\Gamma}^\dagger_{\pi_0,x}\mathbf{q}_{\pi_e,x}]$, $\rho_{PI} := \sup_x \sup_a \left|\mathbf{q}^T_{\pi_e,x}\mathbf{\Gamma}^\dagger_{\pi_0,x}\mathrm{f}(a)\right|$, *and* $\mathbf{q}_{\pi_e,x}, \mathbf{\Gamma}_{\pi_0,x}$ *are defined as in Theorem 1.*

The error bounds for both the IPS and the Similarity estimators are derived following the main idea by Metelli et al. [37]. In particular, they provide a bound for the standard IPS estimator based on a Chebyshev-like inequality using the Rényi divergence [44]. The bound for the PI estimator derives from Theorem 1 by Swaminathan et al. [63] after having taken the bias into account.

We can notice how all the bounds exhibit a similar pattern. Each inequality has a component that depends on the approximation error of the estimator ($d^{supp}(\pi_e||\pi_0)$, ϵ^S, ϵ^{PI}, respectively). In particular cases, the approximation error of such estimators can be zero ($d^{supp}(\pi_e||\pi_0) = 0$ when there is full support; $\epsilon^S = 0$ when the similarity assumption holds; $\epsilon^{PI} = 0$ when there is full support on side information and the linearity assumption holds). Still, in general, we have to consider the error resulting from the possible violation of the assumptions. Furthermore, we can notice that this type of error does not decrease with the sample size n. Therefore, if present, it is an irreducible error.

On the other hand, all the bounds have a remaining component that decreases with the sample size and depends on the difference between the evaluation and logging policies. In particular, the first two inequalities display a *polynomial* concentration (the dependence on n and γ is $\mathcal{O}(\sqrt{1/n\gamma})$). This bound has been proven to be tight in [39] for the IPS estimator. For the third inequality, we have a better behavior due to the *exponential* concentration (when the variance σ^2_{PI} is small the dependence on n and γ is $\mathcal{O}(\sqrt{\ln(1/\gamma)/n})$).

Finally, we notice that the first two inequalities are very similar since the IPS estimator and the Similarity one have a comparable structure, but the former depends on $d_2(\pi_e||\pi_0)$, while the latter on $d_2(\bar{\pi}||\pi_0)$. This indicates how the Similarity estimator has an additional degree of freedom for regulating the bias-variance trade-off. Indeed, by inducing a complex but accurate $\bar{\pi}$, it may decrease the bias factor ϵ^S, while increasing $d_2(\bar{\pi}||\pi_0)$. On the other hand, with a simple model of $\bar{\pi}$, it may reduce the $d_2(\bar{\pi}||\pi_0)$ factor, but it may incur a high bias.

4 Experiments

In this section, we empirically evaluate the considered estimators in an OPE problem with real logged bandit feedback. In particular, we will show how the estimators that exploit side information are competitive in practice with traditional techniques, especially in the presence of deficient support. The code used for the experiments can be found at
`https://github.com/recsyspolimi/neurips-2022-ope-side-info`.

4.1 Setup

We mainly make use of two Python packages: *Open Bandit Pipeline* [49] and *PyIEOE* [50]. These two packages are open source and freely available[1][2]. Open Bandit Pipeline is a library with many state-of-the-art Off-Policy estimators already implemented, while PyIEOE is a collection of scripts for evaluating and comparing OPE estimators.

The dataset that we use is the *Open Bandit Dataset* (OBD), released with Open Bandit Pipeline. OBD contains logged bandit feedback from a real-world application (a large-scale fashion e-commerce platform). There are three campaigns available, namely "ALL", "Men", and "Women". We select the "ALL" campaign. The dimension of the context vector x is 20, and the number of actions $|A| = 80$. Importantly, this dataset contains side information regarding the actions (inside the dataset, called *action context*). The action context has 4 features for each action, among which 3 categorical and one real. We discard the last one and apply one-hot encoding on the first 3. In the end, we get a binary feature vector $\mathrm{f}(a)$ for each a with dimension 40.

4.2 Deficient Support Evaluation

OBD consists of two separate datasets collected with two different policies: one with a uniform random policy $\mathcal{D}_r = \{x_i, a_i, r_i, \pi_r(a_i|x_i) = \frac{1}{|A|}\}_{i=1}^{n_r}$, and the other with a Bernoulli Thompson Sampling (BTS) policy $\mathcal{D}_{\text{bts}} = \{x_i, a_i, r_i, \pi_{\text{bts}}(a_i|x_i)\}_{i=1}^{n_{\text{bts}}}$. Since both policies satisfy the full support assumption, we have to pre-process the data in order to simulate a deficient support scenario.

Specifically, we select the random policy as the logging one ($\pi_0 = \pi_r$), and the BTS as the evaluation policy ($\pi_e = \pi_{\text{bts}}$). Since we have a dataset collected with the BTS, we can easily compute its policy value by on-policy estimation, which we call $R_{\text{on}}(\pi_e) := \frac{1}{n_{\text{bts}}} \sum_{r \in \mathcal{D}_{\text{bts}}} r$. We select a set of random seeds, $\mathcal{S}$. We select a deficient support rate p. This rate represents the percentage of the actions that should have deficient support for each context under the logging policy.

For each random seed $s \in \mathcal{S}$, our pre-processing algorithm proceeds as follows: (i) Select a random sub-sample of $n^* = 100,000$ rows from the logging dataset $\mathcal{D}_r$; (ii) Select p actions randomly for each context, obtaining $\text{Un}(x, \pi_e, \pi_0)$; (iii) For each context x, discard the data points $(x, a, r, \pi_0(a|r))$ where $a \in \text{Un}(x, \pi_e, \pi_0)$; (iv) Modify the logged propensity for each data point: since we know that the logging policy was a uniform random, we need to set $\pi_0(a|x) = 1/|A|(100\% - p)$; (v) Now, we can use the resulting dataset to get the estimate $\hat{R}(\pi_e)$ and get the squared error w.r.t. $R_{\text{on}}(\pi_e)$. This pre-processing protocol is summarized in Algorithm 1, presented in Appendix B.

4.3 Hyperparameter Selection

The proposed Similarity estimator has a weighting function to be set. In the following experimental evaluation, we present two possible variants. The crucial aspect is that they both exploit side information of the actions so that we have a weighting for unsupported actions. Other weighting functions can be proposed, but we leave this analysis as future work.

The first proposal is to have a weighting function proportional to the *cosine* similarity among side information: $w_x(a, a') \propto s_{\cos}(a, a') := f(a)^T f(a')/\|f(a)\|_2 \|f(a')\|_2$. The cosine similarity is typical in fields like recommendation systems [16] and information retrieval [53]. We scale the weighting function such that $\sum_{a' \in A_0(x)} w_x(a, a') = 1$ for any $a \in A_e(x)$ (in accordance with

[1] Open Bandit Pipeline available with Apache License 2.0: `https://github.com/st-tech/zr-obp`

[2] PyIEOE available with MIT License: `https://github.com/sony/pyIEOE`

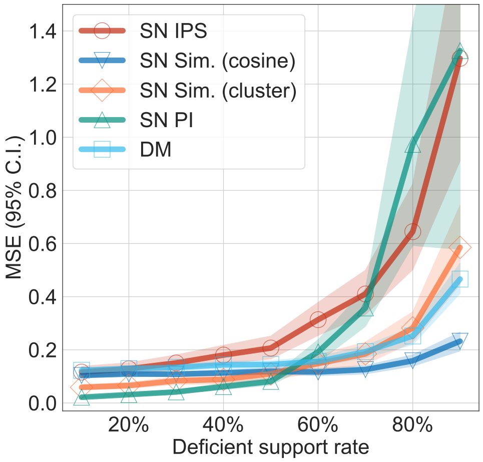


Figure 1: MSE ($\times 10^5$) of the estimators for each deficient support rate.

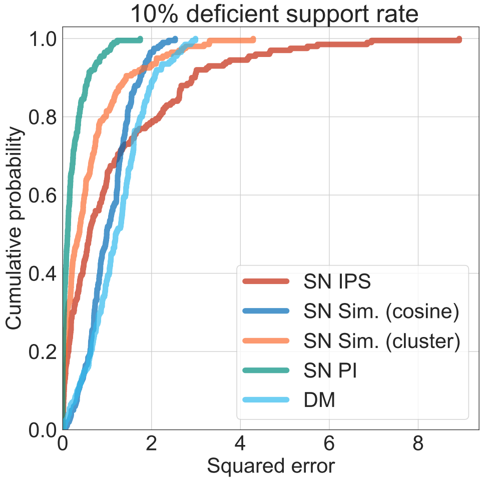


Figure 2: Cumulative Distribution Functions of the squared error ($\times 10^6$) for 10% deficient support rate.

	10%	20%	30%	40%	50%	60%	70%	80%	90%
SN IPS	5.22	3.83	3.50	2.91	2.78	3.43	4.11	5.17	6.72
DM	3.60	2.60	2.14	1.65	1.43	1.32	1.51	1.60	1.73
SN PI	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	2.04	3.39	8.30	7.29
SN Sim. (cosine)	2.96	2.16	1.72	1.28	1.13	**1.00**	**1.00**	**1.00**	**1.00**
SN Sim. (cluster)	2.59	1.95	1.93	1.39	1.47	1.68	1.88	2.24	2.90

Table 1: CVaR$_{0.7}$ with various deficient support rates. Values are normalized for each deficient support rate, the best result is highlighted in bold.

Assumption 4). In this way, the resulting $\bar{\pi}$ satisfies $\sum_{a \in A_0(x)} \bar{\pi}(a|x) = 1$. Hence, the Self-Normalized version of the estimator is consistent under the same assumptions of Theorem 2 [61].

The second proposal is to create a clustering of the actions, which induces a partition of A. This is done by applying *K-Means Clustering* [17, 34] (we set $k = 30$) applied on the normalized action feature vectors $f(a)/\|f(a)\|_2$. Let us define the function $c(a)$ that returns, for each action, its corresponding cluster. Then, we can set the weighting function $w_x(a, a') \propto \mathbf{1}(a \in c(a'))$, which is 1 if both a and a' are in the same cluster, and 0 otherwise. Again, we scale the weighting function such that $\sum_{a' \in A_0(x)} w_x(a, a') = 1$. This type of estimator can also be derived starting from an alternative assumption, as we show in Appendix C.

The choice of the similarity function is fundamental for the proposed algorithm. However, implementing a cross-validation procedure for selecting the similarity is not trivial. We design a possible cross-validation procedure in the presence of deficient support, and we present it in Appendix B.1. The idea is based on simulating deficient support on the logged dataset before selecting the similarity function. Notice that, while it represents a promising direction for future research, this proposal has some limitations, which we discuss in Appendix B.1.

4.4 Results

We repeat our bootstrap evaluation for various deficient support rates: $\{10\%, 20\%, \dots, 90\%\}$, with random seeds $\mathcal{S} = \{1, 2, \dots, 200\}$, for a total of 1800 different combinations. Further details on the infrastructure used and computation time can be found in Appendix B. We compare three estimators (PseudoInverse, Similarity with cosine, Similarity with clustering) with the IPS baseline. We use Self-Normalized variants for each estimator because they outperformed non-normalized ones. As a regression-based baseline, we include the Direct Method (DM) estimator[3]. We conducted experiments with two different regression models: logistic regression and LightGBM [24]. LightGBM consistently outperformed the logistic regression model; thus, we show the results obtained for DM with LightGBM. We compute the Mean Squared Error (MSE) for each deficient support rate by averaging the squared errors obtained with different seeds[4]. Figure 1 compares the MSE of the evaluated estimators, varying the deficient support rate. From Figure 1, we see how the Similarity estimators dominate the IPS and the DM baselines for each deficient support rate. For a low rate (10%), the estimator with cosine similarity has more or less the same error as the IPS. Nonetheless, increasing the deficient support, the error of IPS increases greatly, and the proposed Similarity estimators are clear winners if compared to IPS. The clustering estimator has a lower MSE with respect to the cosine one for low deficient support rates. For high deficient support rates (>50%), it has a higher error than the cosine one but is still lower with respect to IPS. Also, when we have high support deficiency, we notice that the DM estimator is preferable to the clustering one. This shows how the Similarity estimator with clustering displays a hybrid behavior between the vanilla IPS and the cosine Similarity estimator. This finding suggests that whenever we have deficient support with low rates, the clustering estimator is preferable; conversely, the one with cosine similarity performs better with high deficient support rates. Regarding the PseudoInverse, it has the best MSE up until 50% of deficient support rate. Instead, for high deficient support (higher than or equal to 70%), it displays an unstable behavior with a high variance. This effect may be caused by the violation of the full support on side information assumption, which is shown in Table 2, Appendix B.

For a more interpretable evaluation, we compute the Cumulative Distribution Function (CDF) of the squared errors of the estimators for each deficient support rate, as done by, e.g., [48, 61, 68]. The plot of the computed CDF[5] for a deficient support rate of 10% is in Figure 2. This curve sheds light on a noteworthy aspect of the evaluated estimators, as we explain in the following. Looking solely at Figure 1, we see how the MSE of the Similarity estimator with cosine is almost equal to the value of the standard IPS for a deficient support rate of 10%. The MSE, though, is not a desirable metric when we are in a risk-sensitive scenario [8], i.e., whenever we want to minimize the error in the worst case. From Figure 2 we notice that, while having a similar MSE, our Similarity estimator with cosine is highly preferable to IPS in the worst case (IPS has a much longer tail).

For a better evaluation of the worst case, we show the *Conditional Value-at-Risk*$_\alpha$ (CVaR$_\alpha$) metric [1] ($\alpha \in [0, 1)$). This metric computes the average error in the worst $(1 - \alpha) \cdot 100\%$ case. It is widely used in risk-sensitive applications (e.g., in financial portfolio optimization [28, 69]). We set $\alpha = 0.7$ (following [50]), meaning that we evaluate the worst 30% outcomes of the squared error[6]. The results for the various deficient support rates are shown in Table 1. We can see how both Similarity variants outperform the IPS baseline, particularly when we have a strong presence of deficient actions. This indicates that the Similarity estimators are suitable also in risk-sensitive applications with deficient support. The PseudoInverse estimator has an even lower CVaR$_{0.7}$ than Similarity estimators until 50% of deficient support. For higher deficient support rates, it suffers a strong performance degradation. For high deficient support rates. the DM estimator has a lower CVaR$_{0.7}$ than the clustering one, but the best estimator is the Similarity with cosine.

5 Related Work

We focus on the mathematical framework of *contextual bandits* [30, 67] with stochastic rewards, which is a specific instance of the more general *Reinforcement Learning* framework [59]. We address the *Off-Policy Evaluation* (OPE) problem, which consists in evaluating a new policy from data logged by a different policy. This problem traces its roots in causal inference [22, 42], and it is related to the estimation of the average treatment effect [19, 20]. Several Off-Policy estimators have been proposed recently for contextual bandit logged data, as this is a very active research area [13, 14, 50, 57, 58, 62, 65, 68]. One of the most used approaches is the Inverse Propensity Score (IPS) estimator [43, 60, 64], which constitutes the heart of many other proposed state-of-the-art estimators [13, 14, 56, 57, 61, 66, 68]. These estimators all rely on the full support assumption, which

[3] In Appendix B, we include results also for the *Doubly Robust* (DR) estimator [13], which is an estimator with both a regression-based component and a IPS component.

[4] In Appendix B, we show the MSE obtained with different dataset sizes (50,000 and 150,000).

[5] The CDF plots for all the deficient support rates are in the Appendix B.

[6] Alternative choices of α can be found in Appendix B.

we argued is challenging to obtain in a real-world application. Thus, our paper focuses on OPE with deficient support. Despite its relevance in real-world applications, this setting is not very explored in the research literature.

London and Joachims [35] suggested an approach for improved OPE estimators whenever there is a *"new action"* problem, i.e., when the action set is expanded after the logging phase. This problem is a specialization of the deficient support problem because it focuses on actions that the logging policy could not take for any context. Instead, in the deficient support scenario, the action set with no support may be different for different contexts. Therefore, our approach is more general. Furthermore, they focus only on improving regression-based estimators using theoretical tools from domain adaptation [3, 21], while we concentrate on estimators with no regression model.

Sachdeva et al. [47] are the first to address the problem of deficient support with contextual bandit feedback data. In particular, they propose three techniques for learning in this scenario. Their most successful technique is a learning method based on the IPS estimator. This means that they do not rely on regression models, as done in this paper. However, they focus only on Off-Policy Learning. Hence, their method can not be used for evaluating other policies but only for learning one. Our work, instead, proposes solutions for Off-Policy Evaluation. OPE can be the first step in a learning procedure that maximizes the given estimator.

In the estimators proposed in this paper, we use side information about the actions. Within the scope of bandits, the term *"side information"* is often used with different meanings. In principle, the contextual bandit problem was called *"bandit with side information"* [30, 67] to distinguish it from the standard multi-armed bandit problem. In our paper, we differentiate contexts and side information about the actions: a context consists of information given at each round before selecting an action, while the side information about the actions is known to the decision-maker and fixed for each action. This kind of side information is actively exploited in some *online* bandit algorithms, which make the assumption that there is a metric space on the actions with a Lipschitz distance function of the rewards [25, 26, 41, 54, 55]. Our approach, instead, focuses on the *offline* scenario.

For the concentration analysis, we took inspiration from Metelli et al. [37] and their follow-up papers [38, 39], where they analyzed the theoretical properties of the IPS-like estimators using theoretical tools from information theory, such as the Rényi divergence [44].

6 Conclusions

In this paper, we focused on Off-Policy Evaluation, paying special attention to the scenario where the full support assumption is violated. We presented two estimators for a better evaluation by exploiting side information about the actions: one of them was an adaptation of an estimator proposed for a different application domain, while the other is a novel contribution of this paper. We evaluated their theoretical guarantees under alternative assumptions without relying on full support. We also provided finite-sample concentration inequalities, relaxing all the assumptions. The experimental evaluation with real-world data showed that the estimators using side information outperform (both in the average-case and in the worst-case scenario) the traditional approaches (like IPS and regression-based estimators) in the presence of deficient support. For future work, a natural direction to follow is to extend this work to the Off-Policy Learning setting by using the proposed estimators. Furthermore, we presented a novel estimator with hyperparameters to be selected. We proposed a preliminary approach for data-driven hyperparameter selection, but it comes with some limitations. Hence, further research effort in this direction is necessary.

References

[1] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

[2] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397–422, 2002.

[3] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2): 151–175, 2010.

[4] Cesare Bernardis and Paolo Cremonesi. Nfc: a deep and hybrid item-based model for item cold-start recommendation. *User Modeling and User-Adapted Interaction*, pages 1–34, 2021.

[5] Alina Beygelzimer and John Langford. The offset tree for learning with partial labels. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, June 28 - July 1, 2009*, pages 129–138. ACM, 2009.

[6] Aurélien Bibaut, Ivana Malenica, Nikos Vlassis, and Mark J. van der Laan. More efficient off-policy evaluation through regularized targeted learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 654–663. PMLR, 2019.

[7] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250, 2001.

[8] Lorenzo Bisi, Luca Sabbioni, Edoardo Vittori, Matteo Papini, and Marcello Restelli. Risk-averse trust region optimization for reward-volatility reduction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4583–4589. ijcai.org, 2020.

[9] Léon Bottou, Jonas Peters, Joaquin Quiñonero Candela, Denis Xavier Charles, Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Y. Simard, and Ed Snelson. Counterfactual reasoning and learning systems: the example of computational advertising. *J. Mach. Learn. Res.*, 14(1): 3207–3260, 2013.

[10] Cynthia Brandt, Richard Morse, Keri Matthews, Kexin Sun, Aniruddha M. Deshpande, Rohit Gadagkar, Dorothy B. Cohen, Perry L. Miller, and Prakash M. Nadkarni. Metadata-driven creation of data marts from an eav-modeled clinical research database. *Int. J. Medical Informatics*, 65(3):225–241, 2002.

[11] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory - COLT 2008, July 9-12, 2008*, pages 355–366. Omnipress, 2008.

[12] Yashar Deldjoo, Mihai Gabriel Constantin, Bogdan Ionescu, Markus Schedl, and Paolo Cremonesi. MMTF-14K: a multifaceted movie trailer feature dataset for recommendation and retrieval. In *ACM MMSys 2018, June 12-15, 2018*, pages 450–455, 2018.

[13] Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, June 28 - July 2, 2011*, pages 1097–1104. Omnipress, 2011.

[14] Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

[15] Nicolò Felicioni. Enhancing counterfactual evaluation and learning for recommendation systems. In *RecSys '22: Sixteenth ACM Conference on Recommender Systems, September 18 - 23, 2022*, pages 739–741. ACM, 2022.

[16] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. A troubling analysis of reproducibility and progress in recommender systems research. *ACM Trans. Inf. Syst.*, 39(2), January 2021.

[17] Edward W Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769, 1965.

[18] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. Offline A/B testing for recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, February 5-9, 2018*, pages 198–206. ACM, 2018.

[19] Jinyong Hahn. On the role of the propensity score in efficient semiparametric estimation of average treatment effects. *Econometrica*, pages 315–331, 1998.

[20] Keisuke Hirano, Guido W Imbens, and Geert Ridder. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4):1161–1189, 2003.

[21] Hal Daumé III and Daniel Marcu. Domain adaptation for statistical classifiers. *J. Artif. Intell. Res.*, 26:101–126, 2006.

[22] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

[23] Ray Jiang, Sven Gowal, Yuqiu Qian, Timothy A. Mann, and Danilo J. Rezende. Beyond greedy ranking: Slate optimization via list-cvae. In *7th International Conference on Learning Representations, ICLR 2019, May 6-9, 2019*. OpenReview.net, 2019.

[24] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017*, pages 3146–3154, 2017.

[25] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, May 17-20, 2008*, pages 681–690. ACM, 2008.

[26] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and experts in metric spaces. *J. ACM*, 66(4):30:1–30:77, 2019.

[27] Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[28] Pavlo Krokhmal, Jonas Palmquist, and Stanislav Uryasev. Portfolio optimization with conditional value-at-risk objective and constraints. *Journal of risk*, 4:43–68, 2002.

[29] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, ICUIMC 2008, January 31 - February 01, 2008*, pages 208–211. ACM, 2008.

[30] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, December 3-6, 2007*, pages 817–824. Curran Associates, Inc., 2007.

[31] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, February 9-12, 2011*, pages 297–306. ACM, 2011.

[32] Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. Counterfactual estimation and optimization of click metrics in search engines: A case study. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, May 18-22, 2015 - Companion Volume*, pages 929–934. ACM, 2015.

[33] Zun Liu, Xing-Nan Wang, Hui Yu, Jian-Yu Shi, and Wen-Min Dong. Predict multi-type drug–drug interactions in cold start scenario. *BMC bioinformatics*, 23(1):1–13, 2022.

[34] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–136, 1982.

[35] Ben London and Thorsten Joachims. Offline policy evaluation with new arms. In *Offline Reinforcement Learning Workshop at Neural Information Processing Systems*, 2020.

[36] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer, 2011.

[37] Alberto Maria Metelli, Matteo Papini, Francesco Faccio, and Marcello Restelli. Policy optimization via importance sampling. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018*, pages 5447–5459, 2018.

[38] Alberto Maria Metelli, Matteo Papini, Nico Montali, and Marcello Restelli. Importance sampling techniques for policy optimization. *J. Mach. Learn. Res.*, 21:141:1–141:75, 2020.

[39] Alberto Maria Metelli, Alessio Russo, and Marcello Restelli. Subgaussian and differentiable importance sampling for off-policy evaluation and learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021*, pages 8119–8132, 2021.

[40] Huseyin Naci and Alec B O'Connor. Assessing comparative effectiveness of new drugs before approval using prospective network meta-analyses. *Journal of clinical epidemiology*, 66(8): 812–816, 2013.

[41] Sandeep Pandey, Deepak Agarwal, Deepayan Chakrabarti, and Vanja Josifovski. Bandits for taxonomies: A model-based approach. In *Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007,*, pages 216–227. SIAM, 2007.

[42] Judea Pearl. *Causality*. Cambridge university press, 2009.

[43] Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), June 29 - July 2, 2000*, pages 759–766. Morgan Kaufmann, 2000.

[44] Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, volume 4, pages 547–562. University of California Press, 1961.

[45] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW '94, Proceedings of the Conference on Computer Supported Cooperative Work, October 22-26, 1994*, pages 175–186. ACM, 1994.

[46] Francesco Ricci, Lior Rokach, and Bracha Shapira, editors. *Recommender Systems Handbook*. Springer, 2015. ISBN 978-1-4899-7636-9.

[47] Noveen Sachdeva, Yi Su, and Thorsten Joachims. Off-policy bandits with deficient support. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, August 23-27, 2020*, pages 965–975. ACM, 2020.

[48] Yuta Saito and Thorsten Joachims. Counterfactual learning and evaluation for recommender systems: Foundations, implementations, and recent advances. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems, 27 September 2021 - 1 October 2021*, pages 828–830. ACM, 2021.

[49] Yuta Saito, Shunsuke Aihara, Megumi Matsutani, and Yusuke Narita. Open bandit dataset and pipeline: Towards realistic and reproducible off-policy evaluation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021*, 2021.

[50] Yuta Saito, Takuma Udagawa, Haruka Kiyohara, Kazuki Mogi, Yusuke Narita, and Kei Tateno. Evaluating the robustness of off-policy evaluation. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems, 27 September 2021 - 1 October 2021*, pages 114–123. ACM, 2021.

[51] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002*, pages 253–260. ACM, 2002.

[52] Sulthana Shams, Daron Anderson, and Douglas J. Leith. Cluster-based bandits: Fast cold-start for recommender system new users. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, July 11-15, 2021*, pages 1613–1616. ACM, 2021.

[53] Amit Singhal et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24 (4):35–43, 2001.

[54] Aleksandrs Slivkins. Contextual bandits with similarity information. *J. Mach. Learn. Res.*, 15 (1):2533–2568, 2014.

[55] Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi. Learning optimally diverse rankings over large document collections. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010*, pages 983–990. Omnipress, 2010.

[56] Alexander L. Strehl, John Langford, Lihong Li, and Sham M. Kakade. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010*, pages 2217–2225, 2010.

[57] Yi Su, Lequn Wang, Michele Santacatterina, and Thorsten Joachims. CAB: continuous adaptive blending for policy evaluation and learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 6005–6014. PMLR, 2019.

[58] Yi Su, Maria Dimakopoulou, Akshay Krishnamurthy, and Miroslav Dudík. Doubly robust off-policy evaluation with shrinkage. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 9167–9176. PMLR, 2020.

[59] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[60] Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 814–823. JMLR.org, 2015.

[61] Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015*, pages 3231–3239, 2015.

[62] Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *J. Mach. Learn. Res.*, 16:1731–1755, 2015.

[63] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miroslav Dudík, John Langford, Damien Jose, and Imed Zitouni. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017*, pages 3632–3642, 2017.

[64] Philip S. Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015*, pages 3000–3006. AAAI Press, 2015.

[65] Aaron David Tucker and Thorsten Joachims. Variance-optimal augmentation logging for counterfactual evaluation in contextual bandits. *CoRR*, abs/2202.01721, 2022.

[66] Nikos Vlassis, Aurélien Bibaut, Maria Dimakopoulou, and Tony Jebara. On the design of estimators for bandit off-policy evaluation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 6468–6476. PMLR, 2019.

[67] Chih-Chun Wang, Sanjeev R Kulkarni, and H Vincent Poor. Bandit problems with side observations. *IEEE Transactions on Automatic Control*, 50(3):338–355, 2005.

[68] Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudík. Optimal and adaptive off-policy evaluation in contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3589–3597. PMLR, 2017.

[69] Shushang Zhu and Masao Fukushima. Worst-case conditional value-at-risk with application to robust portfolio management. *Operations research*, 57(5):1155–1168, 2009.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See the final part of Section 1
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix and `https://github.com/recsyspolimi/neurips-2022-ope-side-info`
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and Appendix B
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 4 and Appendix B
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix B
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 3
   (b) Did you mention the license of the assets? [Yes] See Section 3
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code available at `https://github.com/recsyspolimi/neurips-2022-ope-side-info`
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Appendix B
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Appendix B
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proofs for Section 3 (Estimators for Off-Policy Evaluation with Side Information)

Theorem 2. *Consider the off-policy evaluation problem where the similarity assumption holds. Then, we can define an unbiased estimator $\hat{R}_S(\pi_e)$ of the expected reward of the evaluation policy as:*

$$\hat{R}_S(\pi_e) := \frac{1}{n}\sum_{i=1}^{n} \frac{\bar{\pi}(a_i|x_i)}{\pi_0(a_i|x_i)} r_i,$$

where $\bar{\pi}(a_i|x_i) := \mathbb{E}_{a\sim\pi_e(\cdot|x_i)}\left[w_{x_i}(a,a_i)\right]$.

Proof. Starting from the similarity assumption, we can rewrite the reward function as follows:

$$\begin{aligned}
\delta(x,a) &= \sum_{a'\in A_0(x)} w_x(a,a')\delta(x,a') && \text{(similarity assumption)}\\
&= \sum_{a'\in A_0(x)} \frac{\pi_0(a'|x)}{\pi_0(a'|x)} w_x(a,a')\delta(x,a')\\
&= \mathop{\mathbb{E}}_{a'\sim\pi_0(\cdot|x)}\left[\frac{w_x(a,a')}{\pi_0(a'|x)}\delta(x,a')\right] && \text{(from the definition of } A_0(x) \text{ and of expected value)}
\end{aligned}$$

Hence, we can empirically estimate the reward as $\hat{\delta}(x_i,a) := \frac{w_{x_i}(a,a_i)}{\pi_0(a_i|x_i)} r_i$.

Now, recall the definition of the expected reward of the evaluation policy:

$$R(\pi_e) = \mathop{\mathbb{E}}_{x\sim p(\cdot)} \mathop{\mathbb{E}}_{a\sim\pi_e(\cdot|x)}[\delta(x,a)] = \mathop{\mathbb{E}}_{x\sim p(\cdot)}\left[\sum_{a\in A_e(x)} \pi_e(a|x)\delta(x,a)\right]$$

If we replace the expected value the empirical mean (which is an unbiased estimator of the expected value) we obtain:

$$\hat{R}(\pi_e) = \frac{1}{n}\sum_{i=1}^{n}\sum_{a\in A_e(x_i)} \pi_e(a|x_i)\delta(x_i,a)$$

This is not useful, because $\delta(x_i,a)$ is in general unknown. Nevertheless, we can replace it with its empirical estimator $\hat{\delta}(x_i,a)$ (defined before) and we finally get an unbiased estimator of the expected reward:

$$\hat{R}_S(\pi_e) = \frac{1}{n}\sum_{i=1}^{n}\sum_{a\in A_e(x_i)} \pi_e(a|x_i)\hat{\delta}(x_i,a) = \frac{1}{n}\sum_{i=1}^{n}\sum_{a\in A_e(x_i)} \pi_e(a|x_i)\frac{w_{x_i}(a,a_i)}{\pi_0(a_i|x_i)} r_i$$

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{r_i}{\pi_0(a_i|x_i)}\sum_{a\in A_e(x_i)} \pi_e(a|x_i) w_{x_i}(a,a_i) = \frac{1}{n}\sum_{i=1}^{n}\frac{r_i}{\pi_0(a_i|x_i)}\mathop{\mathbb{E}}_{a\sim\pi_e(\cdot|x_i)}\left[w_{x_i}(a,a_i)\right]$$

$$:= \frac{1}{n}\sum_{i=1}^{n}\frac{\bar{\pi}(a_i|x_i)}{\pi_0(a_i|x_i)} r_i$$

This derivation proves unbiasedness by construction. An alternative proof can be the following:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{x\sim p(\cdot)\\ a\sim\pi_0(\cdot|x)\\ r\sim p(\cdot|x,a)}}\left[\hat{R}_S(\pi_e)\right] &= \mathop{\mathbb{E}}_{\substack{x\sim p(\cdot)\\ a\sim\pi_0(\cdot|x)\\ r\sim p(\cdot|x,a)}}\left[\frac{\bar{\pi}(a|x)}{\pi_0(a|x)}r\right] && \text{(because data are i.i.d.)}\\
&= \mathop{\mathbb{E}}_{\substack{x\sim p(\cdot)\\ a\sim\pi_0(\cdot|x)}}\left[\frac{\bar{\pi}(a|x)}{\pi_0(a|x)}\delta(x,a)\right] && \text{(by definition of } \delta(x,a))\\
&= \mathop{\mathbb{E}}_{\substack{x\sim p(\cdot)\\ a\sim\pi_0(\cdot|x)}}\left[\frac{\delta(x,a)}{\pi_0(a|x)}\sum_{a'\in A_e(x)}\pi_e(a'|x)w_x(a',a)\right] && \text{(by definition of } \bar{\pi})\\
&= \mathop{\mathbb{E}}_{x\sim p(\cdot)}\left[\sum_{a\in A_0(x)}\delta(x,a)\sum_{a'\in A_e(x)}\pi_e(a'|x)w_x(a',a)\right] && \text{(by definition of expected value)}\\
&= \mathop{\mathbb{E}}_{x\sim p(\cdot)}\left[\sum_{a'\in A_e(x)}\pi_e(a'|x)\sum_{a\in A_0(x)}\delta(x,a)w_x(a',a)\right]\\
&= \mathop{\mathbb{E}}_{x\sim p(\cdot)}\left[\sum_{a'\in A_e(x)}\pi_e(a'|x)\delta(x,a')\right] && \text{(similarity assumption)}\\
&= \mathop{\mathbb{E}}_{\substack{x\sim p(\cdot)\\ a'\sim\pi_e(\cdot|x)}}\left[\delta(x,a')\right]\\
&= R(\pi_e)
\end{aligned}$$

□

In the following Lemma, we show that the Support Divergence bounds the estimation bias of the IPS estimator.

Lemma 1. *Let us consider a generic off-policy evaluation problem with the expected reward function* $\delta(x,a)\in[0,1]$ *and the standard IPS estimator* $\hat{R}_{IPS}(\pi_e)=\frac{1}{n}\sum_{i=1}^{n}\frac{\pi_e(a_i|x_i)}{\pi_0(a_i|x_i)}r_i$*. Then:*

$$\epsilon^{IPS} := \left|\mathbb{E}\left[\hat{R}_{IPS}(\pi_e)\right]-R(\pi_e)\right| \leq d^{supp}(\pi_e||\pi_0)$$

Proof.

$$\begin{aligned}
\left|\mathbb{E}\left[\hat{R}_{\text{IPS}}(\pi_e)\right]-R(\pi_e)\right| &= \left|\mathop{\mathbb{E}}_{\substack{x\sim p(\cdot)\\ a\sim\pi_0(\cdot|x)}}\left[\frac{\pi_e(a|x)}{\pi_0(a|x)}\delta(x,a)\right]-\mathop{\mathbb{E}}_{\substack{x\sim p(\cdot)\\ a\sim\pi_e(\cdot|x)}}\left[\delta(x,a)\right]\right|\\
&= \left|\mathop{\mathbb{E}}_{x\sim p(\cdot)}\left[-\sum_{a\in \text{Un}(x,\pi_e,\pi_0)}\pi_e(a|x)\delta(x,a)\right]\right| && \text{(from Proposition 1)}\\
&= \mathop{\mathbb{E}}_{x\sim p(\cdot)}\left[\sum_{a\in \text{Un}(x,\pi_e,\pi_0)}\pi_e(a|x)\delta(x,a)\right] && \text{(since } \pi_e(a|x)\delta(x,a)\geq 0)\\
&\leq \mathop{\mathbb{E}}_{x\sim p(\cdot)}\left[\sum_{a\in \text{Un}(x,\pi_e,\pi_0)}\pi_e(a|x)\right] && \text{(since } \delta(x,a)\in[0,1])\\
&=: d^{supp}(\pi_e||\pi_0)
\end{aligned}$$

□

Proposition 2. *Let us consider a generic off-policy evaluation problem with side information. Let $\hat{R}_{IPS}$ be the IPS estimator, $\hat{R}_S$ the similarity estimator with weighting function w_x, $\hat{R}_{PI}$ the pseudo-inverse estimator. Then, for any $\gamma \in (0,1)$, the following inequalities hold with probability at least $1-\gamma$:*

$$\left|\hat{R}_{IPS}(\pi_e) - R(\pi_e)\right| \leq d^{supp}(\pi_e||\pi_0) + \sqrt{\frac{d_2(\pi_e||\pi_0)}{n\gamma}},$$

$$\left|\hat{R}_{S}(\pi_e) - R(\pi_e)\right| \leq \epsilon^{S} + \sqrt{\frac{d_2(\bar{\pi}||\pi_0)}{n\gamma}},$$

$$\left|\hat{R}_{PI}(\pi_e) - R(\pi_e)\right| \leq \epsilon^{PI} + \sqrt{\frac{2\sigma^2_{PI}\ln(2/\gamma)}{n}} + \frac{2(\rho_{PI}+1)\ln(2/\gamma)}{3n},$$

where $\bar{\pi}(a|x) := \mathbb{E}_{a'\sim\pi_e(\cdot|x)}[w_x(a',a)]$, $\sigma^2_{PI} := \mathbb{E}_{x\sim p(\cdot)}[\mathbf{q}^T_{\pi_e,x}\mathbf{\Gamma}^{\dagger}_{\pi_0,x}\mathbf{q}_{\pi_e,x}]$, $\rho_{PI} := \sup_x \sup_a \left|\mathbf{q}^T_{\pi_e,x}\mathbf{\Gamma}^{\dagger}_{\pi_0,x}\mathrm{f}(a)\right|$, *and* $\mathbf{q}_{\pi_e,x}, \mathbf{\Gamma}_{\pi_0,x}$ *are defined as in Theorem 1.*

Proof. We start by proving the first inequality regarding the IPS estimator. First, we bound the variance of $\hat{R}_{\text{IPS}}$ with the exponentiated Rényi divergence, as done in Lemma 4.1 by Metelli et al. [37].

$$\begin{aligned}
\text{Var}(\hat{R}_{\text{IPS}}(\pi_e)) &= \frac{1}{n}\text{Var}\left(\frac{\pi_e(a|x)}{\pi_0(a|x)}r\right) && \text{(since data is i.i.d.)}\\
&\leq \frac{1}{n}\,\mathbb{E}\left[\left(\frac{\pi_e(a|x)}{\pi_0(a|x)}r\right)^2\right]\\
&\leq \frac{1}{n}\underset{\substack{x\sim p(\cdot)\\ a\sim\pi_0(\cdot|x)}}{\mathbb{E}}\left[\left(\frac{\pi_e(a|x)}{\pi_0(a|x)}\right)^2\right] && \text{(since } r\in[0,1])\\
&=: \frac{1}{n}d_2(\pi_e||\pi_0)
\end{aligned} \tag{1}$$

Now, we can see that:

$$\begin{aligned}
\left|\hat{R}_{\text{IPS}}(\pi_e) - R(\pi_e)\right| &\leq \left|\mathbb{E}\left[\hat{R}_{\text{IPS}}(\pi_e)\right] - R(\pi_e)\right| + \left|\hat{R}_{\text{IPS}}(\pi_e) - \mathbb{E}\left[\hat{R}_{\text{IPS}}(\pi_e)\right]\right| && \text{(triangle inequality)}\\
&= \epsilon^{\text{IPS}} + \left|\hat{R}_{\text{IPS}}(\pi_e) - \mathbb{E}\left[\hat{R}_{\text{IPS}}(\pi_e)\right]\right| && \text{(by definition of } \epsilon^{\text{IPS}})\\
&\leq d^{supp}(\pi_e||\pi_0) + \left|\hat{R}_{\text{IPS}}(\pi_e) - \mathbb{E}\left[\hat{R}_{\text{IPS}}(\pi_e)\right]\right| && \text{(from Lemma 1)}
\end{aligned} \tag{2}$$

Now, for any $\gamma \in (0,1)$, we can apply Chebyshev inequality to the random variable $\hat{R}_{\text{IPS}}(\pi_e)$. We obtain that, with probability at least $1-\gamma$:

$$\begin{aligned}
\left|\hat{R}_{\text{IPS}}(\pi_e) - \mathbb{E}\left[\hat{R}_{\text{IPS}}(\pi_e)\right]\right| &\leq \sqrt{\frac{\text{Var}(\hat{R}_{\text{IPS}}(\pi_e))}{\gamma}} && \text{(Chebyshev)}\\
&\leq \sqrt{\frac{d_2(\pi_e||\pi_0)}{n\gamma}} && \text{(from Eq. 1)}
\end{aligned}$$

Summing all together with Eq. 2, we obtain the first inequality. The second inequality is obtained with the same procedure. For the third one, it suffices to notice that:

$$\left|\hat{R}_{\text{PI}}(\pi_e) - R(\pi_e)\right| \leq \epsilon^{\text{PI}} + \left|\hat{R}_{\text{PI}}(\pi_e) - \mathbb{E}\left[\hat{R}_{\text{PI}}(\pi_e)\right]\right|$$

by the triangle inequality, and then apply Theorem 1 of [63] to bound $\left|\hat{R}_{\text{PI}}(\pi_e) - \mathbb{E}\left[\hat{R}_{\text{PI}}(\pi_e)\right]\right|$.

□

Definition 1. *Let $x \in X$ be any fixed context. We say that w_x satisfies the identity of indiscernibles if, for any pair $a_1, a_2 \in A_e(x)$, the following condition is satisfied:*

$$(\forall a' \in A_0(x),\ \ w_x(a_1, a') = w_x(a_2, a')) \iff a_1 = a_2.$$

Proposition 3. *Let us define the reward function δ restricted to a fixed context $x \in X$ as $\delta_x : A_e(x) \to [0,1]$, where $\delta_x(a) := \delta(x, a)$. If the similarity assumption holds and w_x satisfies Definition 1, then δ_x is Lipschitz-continuous (with constant 1) with respect to the metric space $(A_e(x), D)$, where the metric D is defined as $D(a_1, a_2) := \sum_{a' \in A_0(x)} |w_x(a_1, a') - w_x(a_2, a')|$*

Proof. First, recall that, because of the similarity assumption, for a given context x:

$$\exists w_x : A_e(x) \times A_0(x) \to \mathbb{R} \text{ such that } \forall a \in A_e(x),\ \delta(x, a) = \sum_{a' \in A_0(x)} w_x(a, a')\delta(x, a')$$

Hence, for any $a_1, a_2 \in A_e(x)$:

$$\begin{aligned}
|\delta_x(a_1) - \delta_x(a_2)| &:= |\delta(x, a_1) - \delta(x, a_2)| \\
&= \left| \sum_{a' \in A_0(x)} w_x(a_1, a')\delta(x, a') - \sum_{a' \in A_0(x)} w_x(a_2, a')\delta(x, a') \right| && \text{(similarity assumption)} \\
&= \left| \sum_{a' \in A_0(x)} (w_x(a_1, a') - w_x(a_2, a'))\delta(x, a') \right| \\
&\leq \sum_{a' \in A_0(x)} |(w_x(a_1, a') - w_x(a_2, a'))\delta(x, a')| && \text{(triangle inequality)} \\
&\leq \sum_{a' \in A_0(x)} |w_x(a_1, a') - w_x(a_2, a')| && \text{(since } \delta(x, a') \in [0,1]\text{)} \\
&:= D(a_1, a_2)
\end{aligned}$$

Now, all we have left to do is to note that D is a metric on $A_e(x)$. This is true if D satisfies the following three properties:

- (P1) Identity of indiscernibles: $D(a_1, a_2) = 0 \iff a_1 = a_2$
  - $D(a_1, a_2)$ is a sum of terms ≥ 0. Hence, the only way for $D(a_1, a_2)$ to be 0 is that $\forall a' \in A_0(x)\ \ |w_x(a_1, a') - w_x(a_2, a')| = 0$. This is verified if and only if $a_1 = a_2$ because w_x satisfies the identity of indiscernibles (Definition 1) by hypothesis.
- (P2) Symmetry: $D(a_1, a_2) = D(a_2, a_1)$
  - Follows from the symmetry of the difference in absolute value.
- (P3) Triangle inequality: $D(a_1, a_3) \leq D(a_1, a_2) + D(a_2, a_3)$
  - $D(a_1, a_3) := \sum_{a' \in A_0(x)} |w_x(a_1, a') - w_x(a_3, a')|\, \delta(x, a') = \sum_{a' \in A_0(x)} |w_x(a_1, a') - w_x(a_2, a') + w_x(a_2, a') - w_x(a_3, a')|\, \delta(x, a') \leq \sum_{a' \in A_0(x)} (|w_x(a_1, a') - w_x(a_2, a')| + |w_x(a_2, a') - w_x(a_3, a')|)\delta(x, a') = D(a_1, a_2) + D(a_2, a_3)$.

□

Corollary 1. *Let us consider the same conditions of Proposition 3 except for the identity of indiscernibles condition on w_x. Now, the same D defined in Proposition 3 is a pseudo-metric. Hence, we say that δ_x is pseudo-Lipschitz-continuous with respect to the pseudo-metric space $(Ae(x), D)$.*

Proof. This follows trivially from the proof of Proposition 3, without proving property P1 of the pseudo-metric D. □

Algorithm 1 Deficient Support Evaluation with Real-World Data

```
Input: Dataset $\mathcal{D}_r$ logged with a random policy, action set $A$, number of bootstrap samples $n^*$, deficient support rate $p$, an estimator to be evaluated $\hat{R}$, an evaluation policy $\pi_e$, on-policy ground-truth of the policy value $R_{\text{on}}(\pi_e)$, a set of random seeds $\mathcal{S}$
Output: A set of squared errors $\mathcal{Z}$
$\mathcal{Z} \leftarrow \emptyset$ (initialize set of results)
for $s \in \mathcal{S}$ do
    $\mathcal{D}^* \leftarrow \text{Bootstrap}(\mathcal{D}_r; s, n^*)$    ▷ sample $n^*$ data points
    $\mathcal{D}^* \leftarrow \text{DiscardDeficientActions}(\mathcal{D}^*; s, p)$    ▷ discard $p$ percent of actions for each context
    $\mathcal{D}^* \leftarrow \text{RebalancePropensity}(\mathcal{D}^*; p)$    ▷ set $\pi_0(a_i|x_i) = 1/|A|(100\% - p)$
    $z' \leftarrow \left(R_{\text{on}}(\pi_e) - \hat{R}(\pi_e; \mathcal{D}^*)\right)^2$    ▷ calculate the squared error of the estimator
    $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{z'\}$
end for
Return $\mathcal{Z}$
```

Def. supp. rate	10%	20%	30%	40%	50%	60%	70%	80%	90%
Unsupp. feat. rate	1.86%	3.97%	6.47%	9.62%	13.70%	19.30%	27.28%	39.30%	59.11%

Table 2: Percentage of unsupported features for varying deficient support rates.

B Experiments

In this section, we provide additional notes and results on the experimental evaluation.

Infrastructure We employed an instance called `c6a.8xlarge` from AWS EC2, with 32 cores and 64GB of RAM for our experiments. The Operating System was Ubuntu 20.04. With a dataset size of 100,000 data points, the baselines (SN IPS, Direct Method, and Doubly Robust) and the two Similarity variants (the one with cosine and the other with clustering) were run in parallel; the total computation time for all the 1800 combinations (200 random seeds and 9 deficient support rates) was about 4 hours. For the PseudoInverse, the total computation time for all the 1800 combinations was approximately 12 hours. Furthermore, two sets of additional experiments were run: one with a dataset size $n^* = 50,000$ and the other with a dataset size $n^* = 150,000$. If we sum the computation time of all the experiments, the total computation time is approximately 48 hours.

Further details on the dataset Open Bandit Dataset (OBD) has been released with Open Bandit Pipeline under CC-BY 4.0 license[7]. OBD contains logged bandit feedback from a Japanese large-scale fashion e-commerce platform called ZOZOTOWN[8]. All user features (that compose the context vector) are anonymized with hash functions.

Further details on the pre-processing phase In Algorithm 1, we can see a schematic explanation of the pre-processing procedure we use for the deficient support evaluation. We use $\mathcal{S} = \{1, \ldots, 200\}$, $p \in \{10\%, \ldots, 90\%\}$. For the experiments reported in the main paper, we set $n^* = 100,000$. In the following, we will also show some additional results obtained with $n^* = 50,000$ and $n^* = 150,000$. In Table 2, we show how there can be unsupported features when we increase the deficient support rate ($n^* = 100,000$). We computed the number of unsupported features, for each deficient support rate, averaging the number of features with 0 probability for each seed in $\mathcal{S}$.

Experimental concentration analysis In Table 3, we show the estimated divergence d_{supp}, varying the deficient support rate. This value is estimated as:

$$\hat{d}_{supp}(\pi_e||\pi_0) = \frac{1}{n}\sum_{i=1}^{n} \sum_{a \in \text{Un}(x_i, \pi_e, \pi_0)} \pi_e(a|x_i)$$

[7] Available at: `https://research.zozo.com/data.html`

[8] `https://zozo.jp/`

Def. supp. rate	10%	20%	30%	40%	50%	60%	70%	80%	90%
$\hat{d}_{supp}(\pi_e\|\|\pi_0)$	9.73%	19.72%	29.88%	39.72%	49.68%	59.81%	69.87%	79.89%	90.00%

Table 3: Estimated support divergence for different deficient support rates ($n^* = 100,000$).

Def. supp. rate	10%	20%	30%	40%	50%	60%	70%	80%	90%
$\hat{d}_2(\pi_e\|\|\pi_0)$	6.13	6.15	6.14	6.21	6.19	6.19	6.20	6.21	6.31
$\hat{d}_2(\bar{\pi}\|\|\pi_0)$ (cosine)	1.44	1.82	2.38	3.23	4.65	7.26	12.89	28.92	114.54
$\hat{d}_2(\bar{\pi}\|\|\pi_0)$ (cluster)	4.14	5.14	6.33	7.76	9.78	12.27	15.68	20.61	27.62

Table 4: Estimated exponentiated Rényi divergence for different deficient support rates ($n^* = 100,000$).

We can notice how $\hat{d}_{supp}(\pi_e||\pi_0)$ is almost equal to the deficient support rate p, which is expected: we hide a percentage p of actions recommended by the logging policy, which is a uniform random policy. Therefore, we will hide approximately p percent of data.

In Table 4, we show the estimated divergence d_2, varying the deficient support rate, for IPS estimator and the Similarity estimators (both with cosine and clustering). This value is estimated as:

$$\hat{d}_2(\pi||\pi_0) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)}\right)^2$$

where π is substituted with π_e, $\bar{\pi}$ (computed with the cosine), $\bar{\pi}$ (computed with clustering). We can see that the value $\hat{d}_2(\pi_e||\pi_0)$ does not change too much while varying the deficient support. This effect is expected since the deficient support adds bias to the IPS estimator, and not variance. Instead, we see a raise of $\hat{d}_2(\bar{\pi}||\pi_0)$ when the deficient support rate increases. This is due to the fact that the Similarity estimators are reducing the high bias of IPS at the cost of some variance.

In Table 5, we show the three estimated values of the bounds given by the concentration inequalities we found in Section 3.3, for $\gamma = 0.05$ (i.e., the inequalities hold with probability $\geq 95\%$). Importantly, we could not estimate the approximation error terms for the similarity estimators ϵ^S since we have no data-dependent bound for them. Conversely, the IPS approximation error can be bounded from $d_{supp}(\pi_e||\pi_0)$, which can be easily estimated from data. Therefore, we use the estimated $\hat{d}_{supp}(\pi_e||\pi_0)$ for the first bound, while the last two bound estimations are optimistic ($\epsilon^S = 0$). Even if it is not fair to compare those three estimated quantities, we can still see how the bias of the IPS takes a dominant role in the computation of its bound.

Additional experiments In Figure 3, we plot the MSE obtained while varying the dataset size. In those plots we included also the *Doubly Robust* (DR) baseline. We notice that, with a small dataset ($n^* = 50,000$), the performance of PI degrades faster with the deficient support rate. Also, we notice how DR has intermediate performance between DM and SN IPS, as expected. In Figure 4 and Figure 5, we can see the plots of the CDFs of the estimators for each deficient support rate ($n^* = 100,000$). From each plot, we clearly notice how the CDF of IPS shows, in the worst case, a worse behavior with respect to the two Similarity estimators. This again indicates that, in a risk-averse scenario, we should prefer the Similarity estimators to IPS. Regarding the PseudoInverse, we have an unstable behavior

Def. supp. rate	10%	20%	30%	40%	50%	60%	70%	80%	90%
$\hat{d}_{supp}(\pi_e\|\|\pi_0) + \sqrt{\frac{\hat{d}_2(\pi_e\|\|\pi_0)}{n\gamma}}$	0.13	0.24	0.34	0.44	0.55	0.65	0.76	0.88	1.01
$\sqrt{\frac{\hat{d}_2(\bar{\pi}\|\|\pi_0)}{n\gamma}}$ (cosine)	0.02	0.02	0.03	0.03	0.04	0.06	0.09	0.17	0.48
$\sqrt{\frac{\hat{d}_2(\bar{\pi}\|\|\pi_0)}{n\gamma}}$ (cluster)	0.03	0.04	0.04	0.05	0.06	0.08	0.10	0.14	0.24

Table 5: Estimated upper bounds for different deficient support rates, with $\gamma = 0.05$ ($n^* = 100,000$).

for high deficient support rates ($\geq 70\%$) . This suggests that we should avoid the PseudoInverse estimator whenever we have a high deficient support rate.

In Tables 6, 7, 8, and 9, we can see the estimators' CVaR values for α of 0.7, 0.8, 0.9, and 0.95, respectively ($n^* = 100,000$). Let us focus on the CVaR$_{0.95}$ (Table 9). In this case, we focus on the average squared error in the worst 5% case. This scenario is of substantial interest because it clearly displays how the Similarity estimator is the best one for a risk-sensitive application. Indeed, looking at Figure 1, we see that the MSE of the PseudoInverse is the lowest one for a 40% deficient support rate. However, when we look at the worst 5% of the outcomes, we should prefer the Similarity estimator with cosine, as exhibited in Table 9.

We ran an additional experiment to investigate the behavior of the PI estimator when we pre-processed the provided features. We used random projection [7] with different numbers of components (10, 15, 20, 25, 30), and we compared the performance of PI applied to the pre-processed features with PI applied to the original 40 features with no pre-processing. We focused on the scenario with a support deficiency rate of 50% and repeated the experiments 50 times with different random seeds, $n^* = 100,000$. In Table 10, we report the MSE for each estimator variant. The results show how the original PI has the best performance, but it is comparable with the results obtained with a random projection with 15 or 20 components.

	10%	20%	30%	40%	50%	60%	70%	80%	90%
SN IPS	$2.91\cdot10^{-6}$	$3.24\cdot10^{-6}$	$3.82\cdot10^{-6}$	$4.72\cdot10^{-6}$	$5.54\cdot10^{-6}$	$8.28\cdot10^{-6}$	$1.11\cdot10^{-5}$	$1.82\cdot10^{-5}$	$3.76\cdot10^{-5}$
DM	$2.01\cdot10^{-6}$	$2.20\cdot10^{-6}$	$2.33\cdot10^{-6}$	$2.67\cdot10^{-6}$	$2.85\cdot10^{-6}$	$3.20\cdot10^{-6}$	$4.09\cdot10^{-6}$	$5.63\cdot10^{-6}$	$9.70\cdot10^{-6}$
DR	$2.16\cdot10^{-6}$	$2.34\cdot10^{-6}$	$2.81\cdot10^{-6}$	$3.17\cdot10^{-6}$	$3.35\cdot10^{-6}$	$5.36\cdot10^{-6}$	$7.08\cdot10^{-6}$	$9.54\cdot10^{-6}$	$1.78\cdot10^{-5}$
SN PI	**$5.57\cdot10^{-7}$**	**$8.45\cdot10^{-7}$**	**$1.09\cdot10^{-6}$**	**$1.62\cdot10^{-6}$**	**$1.99\cdot10^{-6}$**	$4.93\cdot10^{-6}$	$9.16\cdot10^{-6}$	$2.92\cdot10^{-5}$	$4.08\cdot10^{-5}$
SN Sim. (cosine)	$1.65\cdot10^{-6}$	$1.83\cdot10^{-6}$	$1.87\cdot10^{-6}$	$2.07\cdot10^{-6}$	$2.25\cdot10^{-6}$	**$2.42\cdot10^{-6}$**	**$2.70\cdot10^{-6}$**	**$3.52\cdot10^{-6}$**	**$5.60\cdot10^{-6}$**
SN Sim. (cluster)	$1.45\cdot10^{-6}$	$1.65\cdot10^{-6}$	$2.11\cdot10^{-6}$	$2.26\cdot10^{-6}$	$2.94\cdot10^{-6}$	$4.05\cdot10^{-6}$	$5.07\cdot10^{-6}$	$7.88\cdot10^{-6}$	$1.63\cdot10^{-5}$

Table 6: CVaR$_{0.7}$ ($n^* = 100,000$), the best result is highlighted in bold.

	10%	20%	30%	40%	50%	60%	70%	80%	90%
SN IPS	$3.54\cdot10^{-6}$	$4.03\cdot10^{-6}$	$4.65\cdot10^{-6}$	$5.86\cdot10^{-6}$	$6.92\cdot10^{-6}$	$1.04\cdot10^{-5}$	$1.39\cdot10^{-5}$	$2.39\cdot10^{-5}$	$5.17\cdot10^{-5}$
DM	$2.19\cdot10^{-6}$	$2.39\cdot10^{-6}$	$2.55\cdot10^{-6}$	$2.94\cdot10^{-6}$	$3.13\cdot10^{-6}$	$3.67\cdot10^{-6}$	$4.67\cdot10^{-6}$	$6.48\cdot10^{-6}$	$1.09\cdot10^{-5}$
DR	$2.68\cdot10^{-6}$	$2.94\cdot10^{-6}$	$3.52\cdot10^{-6}$	$3.97\cdot10^{-6}$	$4.26\cdot10^{-6}$	$6.80\cdot10^{-6}$	$8.77\cdot10^{-6}$	$1.19\cdot10^{-5}$	$2.22\cdot10^{-5}$
SN PI	**$6.86\cdot10^{-7}$**	**$1.05\cdot10^{-6}$**	**$1.32\cdot10^{-6}$**	**$2.04\cdot10^{-6}$**	**$2.41\cdot10^{-6}$**	$6.14\cdot10^{-6}$	$1.14\cdot10^{-5}$	$4.05\cdot10^{-5}$	$5.86\cdot10^{-5}$
SN Sim. (cosine)	$1.78\cdot10^{-6}$	$2.01\cdot10^{-6}$	$2.04\cdot10^{-6}$	$2.29\cdot10^{-6}$	$2.50\cdot10^{-6}$	**$2.75\cdot10^{-6}$**	**$3.09\cdot10^{-6}$**	**$4.03\cdot10^{-6}$**	**$6.73\cdot10^{-6}$**
SN Sim. (cluster)	$1.78\cdot10^{-6}$	$2.01\cdot10^{-6}$	$2.54\cdot10^{-6}$	$2.81\cdot10^{-6}$	$3.73\cdot10^{-6}$	$5.18\cdot10^{-6}$	$6.51\cdot10^{-6}$	$1.01\cdot10^{-5}$	$2.15\cdot10^{-5}$

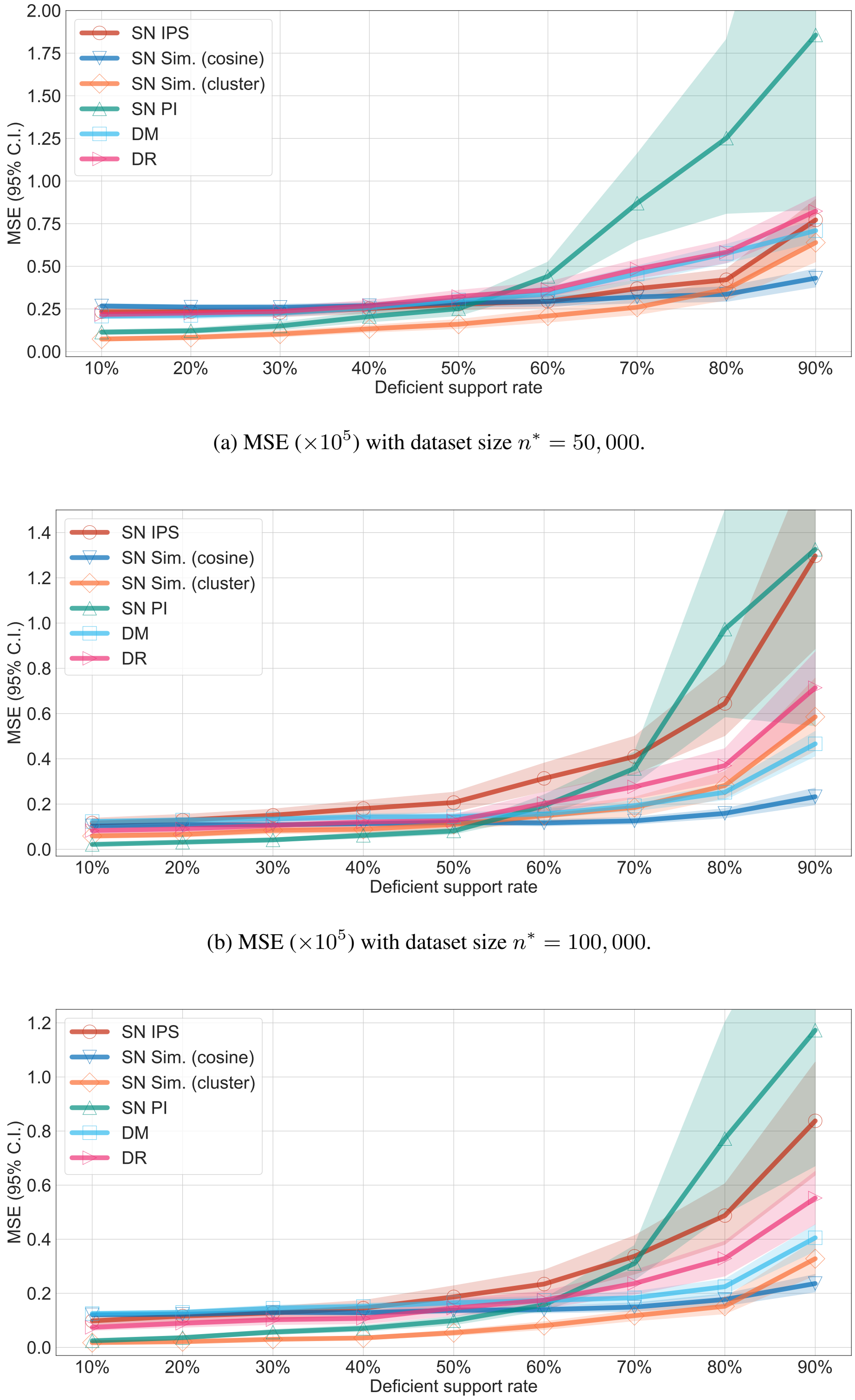
Table 7: CVaR$_{0.8}$ ($n^* = 100,000$), the best result is highlighted in bold.

	10%	20%	30%	40%	50%	60%	70%	80%	90%
SN IPS	$4.53\cdot10^{-6}$	$5.25\cdot10^{-6}$	$6.23\cdot10^{-6}$	$7.76\cdot10^{-6}$	$9.11\cdot10^{-6}$	$1.45\cdot10^{-5}$	$1.88\cdot10^{-5}$	$3.54\cdot10^{-5}$	$8.56\cdot10^{-5}$
DM	$2.48\cdot10^{-6}$	$2.67\cdot10^{-6}$	$2.84\cdot10^{-6}$	$3.31\cdot10^{-6}$	$3.56\cdot10^{-6}$	$4.44\cdot10^{-6}$	$5.46\cdot10^{-6}$	$7.71\cdot10^{-6}$	$1.25\cdot10^{-5}$
DR	$3.55\cdot10^{-6}$	$3.89\cdot10^{-6}$	$5.08\cdot10^{-6}$	$5.58\cdot10^{-6}$	$5.95\cdot10^{-6}$	$9.55\cdot10^{-6}$	$1.21\cdot10^{-5}$	$1.67\cdot10^{-5}$	$3.23\cdot10^{-5}$
SN PI	**$9.06\cdot10^{-7}$**	**$1.36\cdot10^{-6}$**	**$1.67\cdot10^{-6}$**	$2.66\cdot10^{-6}$	$3.22\cdot10^{-6}$	$8.54\cdot10^{-6}$	$1.57\cdot10^{-5}$	$6.91\cdot10^{-5}$	$1.09\cdot10^{-4}$
SN Sim. (cosine)	$1.99\cdot10^{-6}$	$2.23\cdot10^{-6}$	$2.28\cdot10^{-6}$	**$2.63\cdot10^{-6}$**	**$2.90\cdot10^{-6}$**	**$3.34\cdot10^{-6}$**	**$3.60\cdot10^{-6}$**	**$4.97\cdot10^{-6}$**	**$8.48\cdot10^{-6}$**
SN Sim. (cluster)	$2.38\cdot10^{-6}$	$2.56\cdot10^{-6}$	$3.33\cdot10^{-6}$	$3.89\cdot10^{-6}$	$5.32\cdot10^{-6}$	$7.20\cdot10^{-6}$	$9.53\cdot10^{-6}$	$1.35\cdot10^{-5}$	$3.23\cdot10^{-5}$

Table 8: CVaR$_{0.9}$ ($n^* = 100,000$), the best result is highlighted in bold.

	10%	20%	30%	40%	50%	60%	70%	80%	90%
SN IPS	$5.69\cdot10^{-6}$	$6.48\cdot10^{-6}$	$7.97\cdot10^{-6}$	$9.80\cdot10^{-6}$	$1.16\cdot10^{-5}$	$1.91\cdot10^{-5}$	$2.34\cdot10^{-5}$	$4.85\cdot10^{-5}$	$1.26\cdot10^{-4}$
DM	$2.73\cdot10^{-6}$	$2.98\cdot10^{-6}$	$3.10\cdot10^{-6}$	$3.63\cdot10^{-6}$	$3.97\cdot10^{-6}$	$5.04\cdot10^{-6}$	$6.11\cdot10^{-6}$	$8.82\cdot10^{-6}$	$1.41\cdot10^{-5}$
DR	$4.38\cdot10^{-6}$	$4.96\cdot10^{-6}$	$6.74\cdot10^{-6}$	$7.39\cdot10^{-6}$	$7.54\cdot10^{-6}$	$1.32\cdot10^{-5}$	$1.66\cdot10^{-5}$	$2.21\cdot10^{-5}$	$4.76\cdot10^{-5}$
SN PI	**$1.11\cdot10^{-6}$**	**$1.67\cdot10^{-6}$**	**$2.04\cdot10^{-6}$**	$3.34\cdot10^{-6}$	$4.04\cdot10^{-6}$	$1.19\cdot10^{-5}$	$2.10\cdot10^{-5}$	$1.11\cdot10^{-4}$	$2.02\cdot10^{-4}$
SN Sim. (cosine)	$2.15\cdot10^{-6}$	$2.41\cdot10^{-6}$	$2.45\cdot10^{-6}$	**$2.90\cdot10^{-6}$**	**$3.36\cdot10^{-6}$**	**$3.93\cdot10^{-6}$**	**$4.12\cdot10^{-6}$**	**$5.78\cdot10^{-6}$**	**$1.03\cdot10^{-5}$**
SN Sim. (cluster)	$2.92\cdot10^{-6}$	$3.16\cdot10^{-6}$	$4.39\cdot10^{-6}$	$5.14\cdot10^{-6}$	$6.93\cdot10^{-6}$	$8.83\cdot10^{-6}$	$1.28\cdot10^{-5}$	$1.60\cdot10^{-5}$	$4.37\cdot10^{-5}$

Table 9: CVaR$_{0.95}$ ($n^* = 100,000$), the best result is highlighted in bold.



(a) MSE ($\times 10^5$) with dataset size $n^* = 50,000$.

(b) MSE ($\times 10^5$) with dataset size $n^* = 100,000$.

(c) MSE ($\times 10^5$) with dataset size $n^* = 150,000$.

Figure 3: MSE ($\times 10^5$) varying dataset sizes.

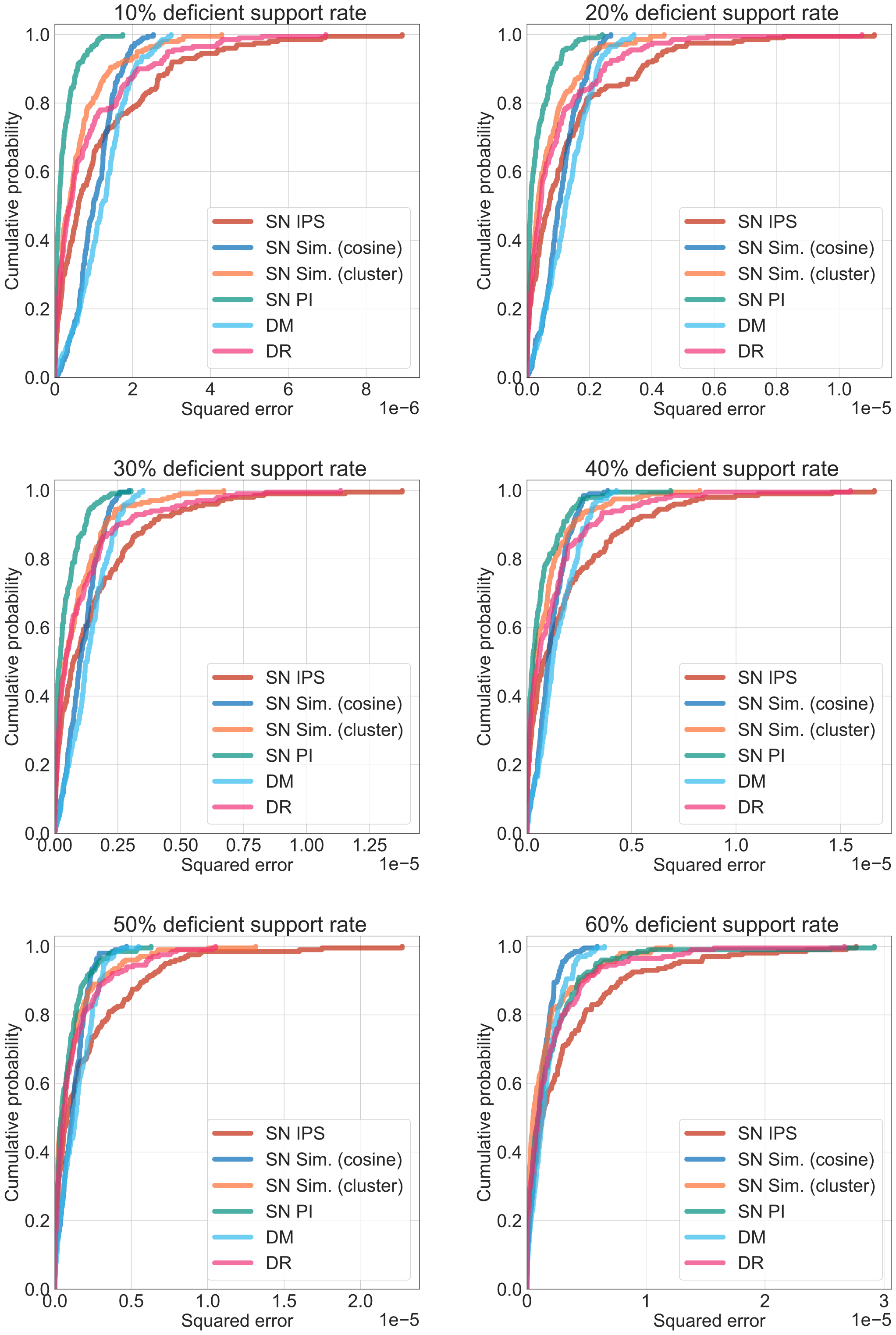


Figure 4: Cumulative Distribution Functions of the squared error varying deficient support rates ($n^* = 100,000$).

Original	10	15	20	25	30
$6.56 \cdot 10^{-7}$	$1.09 \cdot 10^{-6}$	$6.91 \cdot 10^{-7}$	$7.59 \cdot 10^{-7}$	$1.70 \cdot 10^{-6}$	$8.72 \cdot 10^{-7}$

Table 10: Comparison of the MSEs of PI estimators applied to the original features and to preprocessed features.

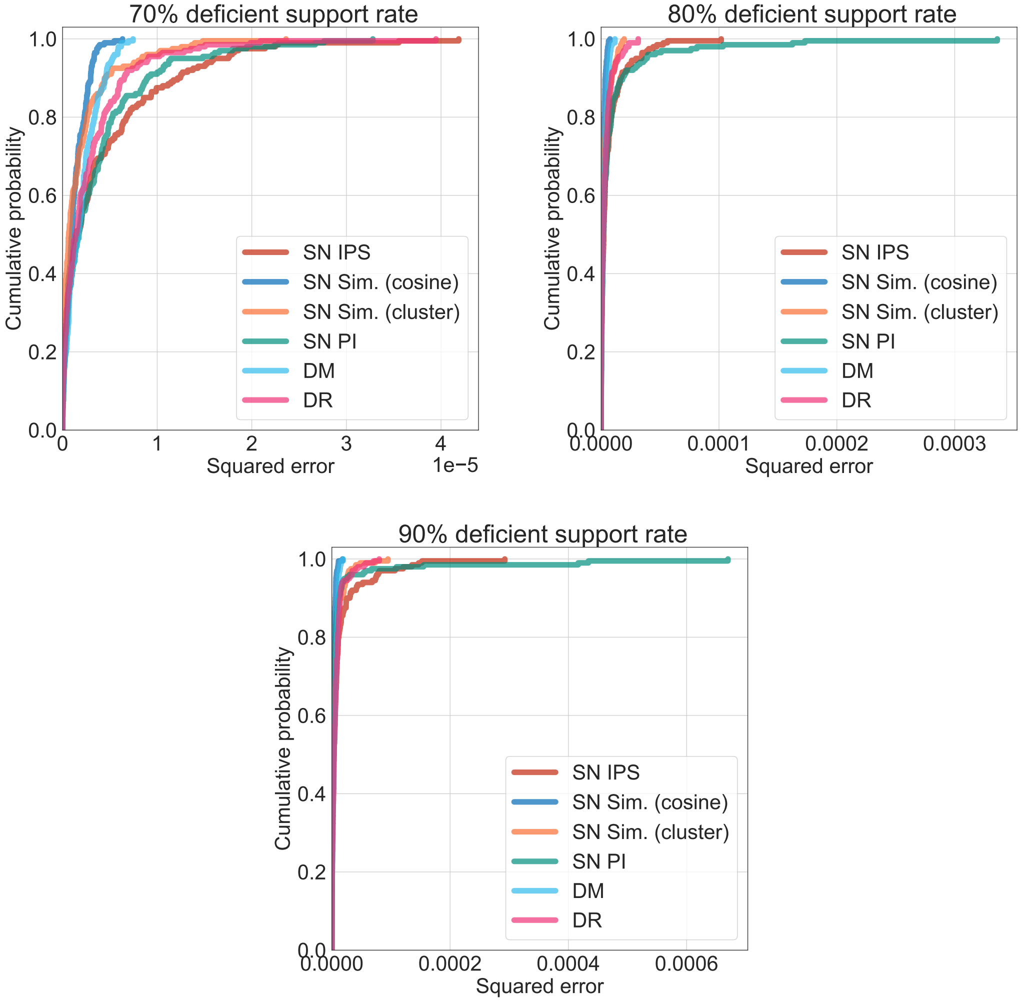


Figure 5: Cumulative Distribution Functions of the squared error varying deficient support rates ($n^{*} = 100,000$).

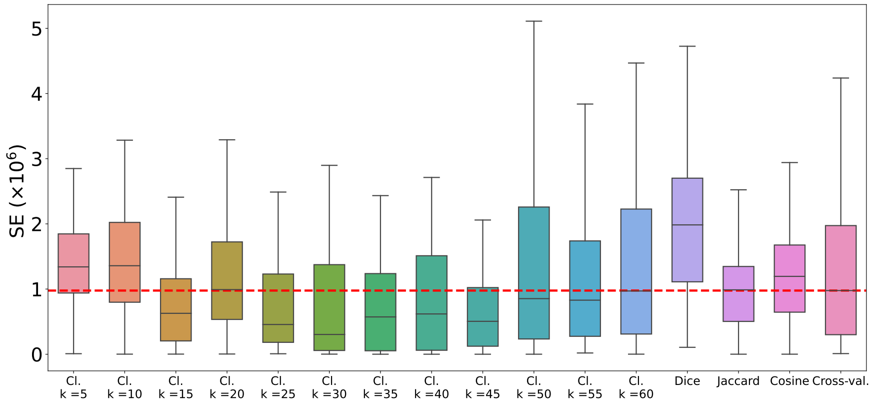


Figure 6: Box plot of the squared errors ($\times 10^6$) obtained by the various similarity estimators with a fixed similarity, compared to the proposed cross-validation procedure. The dashed red line is the median result obtained with the cross-validation procedure.

B.1 Cross-Validation Proposal

In the following, we design a possible cross-validation procedure in the presence of deficient support.

In the Off-Policy Evaluation problem, we have access to a logged dataset $\mathcal{D} = \{x_i, a_i, \pi_0(a_i|x_i), r_i\}_{i=1}^n$, and we want to evaluate the policy value of a different policy, which we will call *evaluation policy* π_e. From such a dataset, we can compute an unbiased estimate of the logging policy π_0 via Monte-Carlo on-policy estimation: $\hat{R}_{on}(\pi_0) = \frac{1}{n}\sum_{i=1}^n r_i$. Let us assume that we know the deficient support rate of π_0 with respect to π_e. We call the deficient support rate p. We can create another dataset $\mathcal{D}'$ by hiding the p percent of actions from $\mathcal{D}$ for each context x_i. In order to simulate the deficient scenario, we should also re-balance the logging policy accordingly. We will call the resulting simulated logging policy π_0'. Now, we can interpret the original logging policy π_0 as the evaluation policy (and we know its ground-truth $\hat{R}_{on}(\pi_0)$), and the new π_0' as the logging policy. In this way, we can try different variants of the similarity estimator (e.g., varying similarity functions) and evaluate those variants with the error with respect to $\hat{R}_{on}(\pi_0)$. Ultimately, we can choose the best similarity according to which one has the lowest cross-validation error.

Limitations We should notice that this approach has two limitations: first, the simulated evaluation policy π_0 and the simulated logging policy π_0' will be very similar. This means that the estimator's validation performance may be optimistic due to low variance. Second, this approach is viable only whenever we have complete knowledge of π_0 (and not only of the propensities for the logged actions $\pi_0(a_i|x_i)$). Without this knowledge, we can not compute the simulated logging policy π_0'. To the best of our knowledge, this is the first proposal of a procedure for OPE hyperparameter selection with deficient support. Therefore, it is a research question that needs to address non-trivial issues and requires a broader discussion that could constitute future work.

Empirical Validation We ran an additional experiment to validate this procedure empirically. We tried to select the similarity function of the similarity estimator among the following possible choices: Dice, Jaccard, Cosine, and Clustering. For the clustering similarity, we tried different values of k (5, 10, . . . , 60). We ran this experiment for a dataset with a size of 100,000 and 50% of support deficiency. The results are the squared errors obtained over 50 random seeds. We compare the result obtained following the proposed cross-validation procedure with the results obtained by fixing one of the candidate similarities. For each random seed, we may have a different logged dataset. Hence, for each random seed, we repeated the validation procedure 5 times (i.e., we created a simulated logging policy 5 times), and we selected the best similarity according to the best average validation error. Notice how this implies that the cross-validation procedure can possibly choose a different similarity function for each seed. The squared errors obtained over 50 random seeds are summarized in the box plot in Figure 6.

As expected, the cross-validation procedure displays intermediate results among the possible choices. If we focus on the median result, we see that:

- It performs better than fixing the Dice similarity or the Cosine one. It also outperforms clustering similarities with $k \leq 10$.
- It has comparable performance with the clustering with $k \in \{20, 50, 55, 60\}$ and with the Jaccard similarity.
- It is outperformed by some other similarities, for instance, by the clustering ones with $25 \leq k \leq 45$.

C Alternative Derivation for Similarity Estimator with Clustering

In this section, we derive an alternative estimator, which we initially call *Clustering* estimator. In Lemma 2, we show that this is actually a special case of the Similarity estimator, with a particular choice of the weighting function. The derivation is based on the following assumption:

Assumption 5 (Clustering)**.** *The off-policy evaluation problem satisfies the clustering assumption if, for any context x, there exists a partition (called clustering) $C(x) = (c_1, c_2, \dots, c_k)$ of $A_e(x)$, such that, for any two actions a, a', if they belong to the same cluster ($a, a' \in c$), then, the reward distribution satisfies the following condition: $p(\cdot|x, a) = p(\cdot|x, a') = p(\cdot|x, c)$.*

For simplicity, let us define the cluster membership function as $c(\cdot|x) : A_e(x) \to C(x)$ that returns, for each action, the corresponding cluster. We can view this assumption as the fact that we may have similar actions (which we group into the same cluster) such that the expected reward for those actions is the same: $\delta(x, a_i) = \delta(x, a_j) = \delta(x, c)$ if $c(a_i|x) = c(a_j|x) = c$. In order to derive an unbiased estimator, we also need a full-support assumption on the clustering space.

Assumption 6 (Full Support on Clustering)**.** *Let us consider an off-policy evaluation problem that satisfies the clustering assumption. The OPE problem satisfies the full support assumption on the clustering if, for any cluster $c \in C(x)$, there is at least one action $a \in c$ that is supported by the logging policy $a \in A_0(x)$, with probability one over $x \sim p(\cdot)$.*

This assumption is less demanding than the standard full support because we only need at least one supported action per cluster.

The following Lemma shows how the Clustering estimator is actually a Similarity estimator with a particular selection of the weighting function.

Lemma 2. *Let us consider an off-policy evaluation problem that satisfies the clustering assumption and has full support on clustering. Then, the off-policy problem satisfies also the similarity assumption, with w_x defined as follows:*

$$w_x(a, a') = \frac{\mathbf{1}(a' \in c(a|x))}{\sum_{a_0 \in A_0(x)} \mathbf{1}(a_0 \in c(a|x))}$$

Proof. From the clustering assumption, it follows that $\delta(x, a) = \delta(x, c(a|x))$ for any a. Hence:

$$\begin{aligned} \delta(x, a) &= \delta(x, a) \cdot \frac{\sum_{a_0 \in A_0(x)} \mathbf{1}(a_0 \in c(a|x))}{\sum_{a_0 \in A_0(x)} \mathbf{1}(a_0 \in c(a|x))} && \text{(from the full support on clustering there is at least one } a_0 \in c(a|x)\text{)} \\ &= \frac{\sum_{a_0 \in A_0(x)} \mathbf{1}(a_0 \in c(a|x))\delta(x, a_0)}{\sum_{a_0 \in A_0(x)} \mathbf{1}(a_0 \in c(a|x))} && \text{(because of the clustering assumption, } \delta(x, a_0) = \delta(x, a) \text{ whenever } a_0 \in c(a|x)\text{)} \\ &:= \sum_{a_0 \in A_0(x)} w_x(a, a_0)\delta(x, a_0) \end{aligned}$$

□

Now, we can present an unbiased estimator for this setting.

Theorem 3. *Consider the off-policy evaluation problem where Assumption 5 and Assumption 6 hold. Then, we can define an unbiased estimator $\hat{R}_{CL}(\pi_e)$ of the expected reward of the evaluation policy as:*

$$\hat{R}_{CL}(\pi_e) := \frac{1}{n} \sum_{i=1}^{n} \frac{\pi_e(c_i|x_i)}{\pi_0(a_i|x_i)} \frac{r_i}{N(x_i, c_i)}$$

where $c_i := c(a_i|x_i)$, $\pi_e(c|x) := \sum_{a \in c} \pi_e(a|x)$, $N(x, c) := \sum_{a' \in A_0(x)} \mathbf{1}(a' \in c)$.

Proof. We start from Lemma 2 and we rewrite the formulation of $\delta(x,a)$ for each $x \in X$ and for each $a \in A_e(x)$.

$$\begin{aligned}
\delta(x,a) &= \frac{\sum_{a' \in A_0(x)} \mathbf{1}(a' \in c(a|x))\delta(x,a')}{\sum_{a' \in A_0(x)} \mathbf{1}(a' \in c(a|x))} && \text{(from Lemma 2)} \\
&= \frac{1}{N(x,c(a|x))} \sum_{a' \in A_0(x)} \mathbf{1}(a' \in c(a|x))\delta(x,a') && \text{(by definition of } N(x,c(a|x))\text{)} \\
&= \frac{1}{N(x,c(a|x))} \sum_{a' \in A_0(x)} \mathbf{1}(a' \in c(a|x))\delta(x,a')\frac{\pi_0(a'|x)}{\pi_0(a'|x)} \\
&= \frac{1}{N(x,c(a|x))} \mathop{\mathbb{E}}_{a' \sim \pi_0(\cdot|x)} \left[\frac{\mathbf{1}(a' \in c(a|x))}{\pi_0(a'|x)} \delta(x,a') \right]
\end{aligned}$$

Now, we can proceed in a similar way as Theorem 2. First, we create an unbiased estimator of the reward by taking a single sample:

$$\hat{\delta}(x_i,a) := \frac{1}{N(x_i,c(a|x))} \frac{\mathbf{1}(a_i \in c(a|x_i))}{\pi_0(a_i|x_i)} r_i$$

Then, we plug $\hat{\delta}(x_i,a)$ inside the empirical mean estimator of the reward of π_e:

$$\begin{aligned}
\hat{R}_{CL}(\pi_e) &= \frac{1}{n}\sum_{i=1}^{n} \sum_{a \in A_e(x_i)} \pi_e(a|x_i)\hat{\delta}(x_i,a) \\
&= \frac{1}{n}\sum_{i=1}^{n} \sum_{a \in A_e(x_i)} \pi_e(a|x_i) \frac{1}{N(x_i,c(a|x))} \frac{\mathbf{1}(a_i \in c(a|x_i))}{\pi_0(a_i|x_i)} r_i \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{r_i}{\pi_0(a_i|x_i)} \sum_{a \in A_e(x_i)} \pi_e(a|x_i) \frac{\mathbf{1}(a_i \in c(a|x_i))}{N(x_i,c(a|x))}
\end{aligned}$$

Now, we notice that the second summation takes non-zero values only when $a_i \in c(a|x_i)$, i.e., when $c(a|x_i) = c(a_i|x_i) := c_i$. Hence, we can rewrite the sum as:

$$\begin{aligned}
\hat{R}_{CL}(\pi_e) &= \frac{1}{n}\sum_{i=1}^{n} \frac{r_i}{\pi_0(a_i|x_i)} \sum_{a \in A_e(x_i)} \pi_e(a|x_i) \frac{\mathbf{1}(a_i \in c(a|x_i))}{N(x_i,c(a|x))} \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{r_i}{\pi_0(a_i|x_i)} \sum_{a \in c_i} \frac{\pi_e(a|x_i)}{N(x_i,c_i)} \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{r_i}{\pi_0(a_i|x_i)} \frac{1}{N(x_i,c_i)} \sum_{a \in c_i} \pi_e(a|x_i) \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{\pi_e(c_i|x_i)}{\pi_0(a_i|x_i)} \frac{r_i}{N(x_i,c_i)} && \text{(by definition of } \pi_e(c_i|x_i)\text{)}
\end{aligned}$$

□