

Cross-Validated Off-Policy Evaluation

Matej Cief^{1,2}, Branislav Kveton³, Michal Kompan²

¹Brno University of Technology

²Kempelen Institute of Intelligent Technologies

³Adobe Research*

Abstract

We study estimator selection and hyper-parameter tuning in off-policy evaluation. Although cross-validation is the most popular method for model selection in supervised learning, off-policy evaluation relies mostly on theory, which provides only limited guidance to practitioners. We show how to use cross-validation for off-policy evaluation. This challenges a popular belief that cross-validation in off-policy evaluation is not feasible. We evaluate our method empirically and show that it addresses a variety of use cases.

Code — <https://github.com/navarog/cross-validated-ope>

Extended version — <https://arxiv.org/abs/2405.15332>

1 Introduction

Off-policy evaluation (OPE, Li et al. 2010) is a framework for estimating the performance of a policy without deploying it online. It is useful in domains where online A/B testing is costly or too dangerous. For example, deploying an untested algorithm in recommender systems or advertising can lead to a loss of revenue (Li et al. 2010; Silver et al. 2013), and in medical treatments, it may have a detrimental effect on the patient’s health (Hauskrecht and Fraser 2000). A popular approach to off-policy evaluation is *inverse propensity scoring* (IPS, Robins, Rotnitzky, and Zhao 1994). As this method is *unbiased*, it approaches a true policy value with more data.

However, when the data logging policy has a low probability of choosing some actions, IPS-based estimates have a high *variance* and often require a large amount of data to be useful in practice (Dudik et al. 2014). Therefore, other lower-variance methods have emerged. These methods often have hyper-parameters, such as a clipping constant to truncate large propensity weights (Ionides 2008). Some works provide theoretical insights (Ionides 2008; Metelli, Russo, and Restelli 2021) for choosing hyper-parameters, while there are none for many others.

In supervised learning, data-driven techniques for hyper-parameter tuning, such as cross-validation, are more popular than theory-based techniques, such as the Akaike information criterion (Bishop 2006). The reason is that they perform

better on large datasets, which are standard today. Unlike in supervised learning, the ground truth value of the target policy is unknown in off-policy evaluation. A common assumption is that standard machine learning approaches for model selection would fail because there is no unbiased and low-variance approach to compare estimators (Su, Srinath, and Krishnamurthy 2020). Therefore, only a few works studied estimator selection for off-policy evaluation, and no general solution exists (Saito et al. 2021; Udagawa et al. 2023).

Despite common beliefs, we show that cross-validation in off-policy evaluation can be done comparably to supervised learning. In supervised learning, we do not know the true data distribution, but we are given samples from it. Each sample is an unbiased and high-variance representation of this distribution. Nevertheless, we can still get an accurate estimate of true generalization when averaging the model error over these samples in cross-validation. Similarly, we do not know the true reward distribution in off-policy evaluation, but we are given high-variance samples from it. The difference is that these samples are biased because they are collected by a different policy. However, we can use an unbiased estimator, such as IPS, on a held-out validation set to get an unbiased estimate of any policy value. Then, as with supervised learning, we get an estimate of the estimator’s performance. Our contributions are:

- We propose an easy-to-use estimator selection procedure for off-policy evaluation based on cross-validation that requires only data collected by a single policy.
- We analyze the loss of our procedure and how it relates to the true loss if the ground truth policy value was known. We use this insight to reduce its variance.
- We empirically evaluate the procedure on estimator selection and hyper-parameter tuning problems using nine real-world datasets. The procedure is more accurate than prior techniques and computationally efficient.

2 Off-Policy Evaluation

A contextual bandit (Langford, Strehl, and Wortman 2008) is a popular model of an agent interacting with an unknown environment. The interaction in round i starts with the agent observing a *context* $x_i \in \mathcal{X}$, which is drawn i.i.d. from an unknown distribution p , where \mathcal{X} is the *context set*. Then the agent takes an *action* $a_i \sim \pi(\cdot | x_i)$ from the *action set*

\mathcal{A} according to its policy π . Finally, it receives a stochastic reward $r_i = r(x_i, a_i) + \varepsilon_i$, where $r(x, a)$ is the mean reward of action a in context x and ε_i is an independent zero-mean noise.

In *off-policy evaluation* (Li et al. 2010), a *logging policy* π_0 interacts with the bandit for n rounds and collects a *logged dataset* $\mathcal{D} = \{(x_i, a_i, r_i)\}_{i=1}^n$. The goal is to estimate the value of a *target policy*

$$V(\pi) = \sum_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}} p(x) \pi(a | x) r(x, a)$$

using the dataset \mathcal{D} . Various estimators have been proposed to either correct for the distribution shift caused by the differences in π and π_0 , or to estimate $r(x, a)$. We review the canonical ones below and leave the rest to Appendix A.

The *inverse propensity scores* estimator (IPS, Robins, Rotnitzky, and Zhao 1994) reweights logged samples as if collected by the target policy π ,

$$\hat{V}_{\text{IPS}}(\pi; \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n \frac{\pi(a_i | x_i)}{\pi_0(a_i | x_i)} r_i. \quad (1)$$

This estimator is unbiased but suffers from a high variance. Therefore, a clipping constant is often used to truncate high propensity weights (Ionides 2008). This is a hyper-parameter that needs to be tuned.

The *direct method* (DM, Dudik et al. 2014) is a popular approach to off-policy evaluation. Using the DM, the policy value estimate can be computed as

$$\hat{V}_{\text{DM}}(\pi; \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n \sum_{a \in \mathcal{A}} \pi(a | x_i) \hat{f}(x_i, a), \quad (2)$$

where $\hat{f}(x, a)$ is an estimate of the mean reward $r(x, a)$ from \mathcal{D} . The function \hat{f} is chosen from some function class, such as linear functions.

The *doubly-robust* estimator (DR, Dudik et al. 2014) combines the DM and IPS as

$$\hat{V}_{\text{DR}}(\pi; \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n \frac{\pi(a_i | x_i)}{\pi_0(a_i | x_i)} (r_i - \hat{f}(x_i, a_i)) + \hat{V}_{\text{DM}}(\pi; \mathcal{D}), \quad (3)$$

where $\hat{f}(x, a)$ is an estimate of $r(x, a)$ from \mathcal{D} . The DR is unbiased when the DM is, or the propensity weights are correctly specified. The estimator is popular in practice because $r_i - \hat{f}(x_i, a_i)$ reduces the variance of rewards in the IPS part of the estimator.

Many other estimators with tunable parameters exist: TruncatedIPS (Ionides 2008), SWITCH-DR (Wang, Agarwal, and Dudik 2017), Continuous OPE (Kallus and Zhou 2018), CAB (Su et al. 2019), DRos and DRps (Su et al. 2020), IPS- λ (Metelli, Russo, and Restelli 2021), MIPS (Saito and Joachims 2022), Exponentially smooth IPS (Aouali et al. 2023), GroupIPS (Peng et al. 2023), OffCEM (Saito, Ren, and Joachims 2023), Policy Convolution (Sachdeva et al. 2024), Learned MIPS (Cief et al. 2024), and subtracting control variates (Vlassis et al. 2019). Some of these works leave

the hyper-parameter selection as an open problem, while others provide a theory for selecting an optimal hyper-parameter, usually by bounding the bias of the estimator. As in supervised learning, we show that theory is often too conservative, and given enough data, our method can select better hyper-parameters. Other works use the statistical Lepski’s adaptation method (Lepski and Spokoiny 1997), which requires that the hyper-parameters are ordered so that the bias is monotonically increasing. The practitioner also needs to choose the estimator. To address these shortcomings, we adapt cross-validation, a well-known machine learning technique for model selection, to estimator selection in a way that is general and applicable to *any* estimator.

3 Related Work

To the best of our knowledge, there are only a few data-driven approaches for estimator selection or hyper-parameter tuning in off-policy evaluation for bandits. We review them below.

Su, Srinath, and Krishnamurthy (2020) propose a hyper-parameter tuning method SLOPE based on Lepski’s principle (Lepski and Spokoiny 1997). The key idea is to order the hyper-parameter values so that the estimators’ variances decrease. Then, we compute the confidence intervals for all the values in this order. If a confidence interval does not overlap with *all* previous intervals, we stop and select the previous value. While the method is straightforward, it assumes that the hyper-parameters are ordered such that the bias is monotonically increasing. This makes it impractical for estimator selection, where it may be difficult to establish a correct order of the estimators.

Saito et al. (2021) rely on a logged dataset collected by multiple logging policies. They use one of the logging policies as the pseudo-target policy and directly estimate its value from the dataset. Then, they choose the off-policy estimator that most accurately estimates the pseudo-target policy. This approach assumes that we have access to a logged dataset collected by multiple policies. Moreover, it ultimately chooses the best estimator for the pseudo-target policy, and not the target policy. Prior empirical studies (Voloshin et al. 2021) showed that the estimator’s accuracy greatly varies when applied to different target policies.

In PAS-IF (Udagawa et al. 2023), two new surrogate policies are created using the logged dataset. The surrogate policies have two properties: 1) the propensity weights from surrogate logging and target policies imitate those of the true logging and target policies, and 2) the logged dataset can be split in two as if each part was collected by one of the surrogate policies. They learn a neural network that optimizes this objective. Then, they evaluate estimators as in Saito et al. (2021), using surrogate policies and a precisely split dataset. They show that estimator selection on these surrogate policies adapts better to the true target policy.

In this work, we do not require multiple logging policies, make no strong assumptions, and use principal techniques from supervised learning that are well-known and loved by practitioners. Therefore, our method is easy to implement and, as showed in Section 6, also more accurate.

A popular approach in offline policy selection (Lee et al. 2022; Nie et al. 2022; Saito and Nomura 2024) is to evalu-

ate candidate policies on a held-out set by OPE. Nie et al. (2022) even studied a similar approach to cross-validation. While these papers seem similar to our work, the problems are completely different. All estimators in our work estimate the same value $V(\pi)$, and this structure is used in the design of our solution (Section 5). We also address important questions that the prior works did not, such as how to choose a validator and how to choose the training-validation split. A naive application of cross-validation without addressing these issues fails in OPE (Appendix B).

4 Cross-Validation in Machine Learning

Model selection (Bishop 2006) is a classic machine learning problem. It can be addressed by two kinds of methods. The first approach is probabilistic model selection, such as the Akaike information criterion (Akaike 1998) and Bayesian information criterion (Schwarz 1978). These methods penalize the complexity of the learned model during training (Stoica and Selen 2004). They are designed using theory and do not require a validation set. Broadly speaking, they work well on smaller datasets because they favor simple models (Bishop 2006). The second approach estimates the performance of models on a held-out validation set, such as *cross-validation* (CV, Stone 1974). CV is a state-of-the-art approach for large datasets and neural networks (Yao, Rosasco, and Caponnetto 2007). We focus on this setting because large amounts of logged data are available in modern machine learning.

In the rest of this section, we introduce cross-validation. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a function that maps features $x \in \mathbb{R}^d$ to \mathbb{R} . It belongs to a function class \mathcal{F} . For example, f is a linear function, and \mathcal{F} is the class of linear functions. A machine learning algorithm Alg maps a dataset \mathcal{D} to a function in \mathcal{F} . We write this as $f = \text{Alg}(\mathcal{F}, \mathcal{D})$. One approach to choosing f is to minimize the *squared loss* on \mathcal{D} ,

$$L(f, \mathcal{D}) = \sum_{(x,y) \in \mathcal{D}} (y - f(x))^2,$$

which can be written as

$$\text{Alg}(\mathcal{F}, \mathcal{D}) = \arg \min_{f \in \mathcal{F}} L(f, \mathcal{D}). \quad (4)$$

This leads to overfitting on \mathcal{D} (Devroye, Györfi, and Lugosi 1996). To prevent this, CV is commonly used to evaluate f on unseen validation data to give a more honest estimate of its generalization ability. In K -fold CV, the dataset is split into K folds. We denote the validation data in the k -th fold by $\tilde{\mathcal{D}}_k$ and all other training data by $\hat{\mathcal{D}}_k$. Using this notation, the average loss on a held-out set can be formally written as $\frac{1}{K} \sum_{k=1}^K L(\text{Alg}(\mathcal{F}, \hat{\mathcal{D}}_k), \tilde{\mathcal{D}}_k)$.

Cross-validation can be used to select a model as follows. Suppose that we have a set of function classes $\mathbf{F} = \{\mathcal{F}\}$. For instance, $\mathbf{F} = \{\mathcal{F}_1, \mathcal{F}_2\}$, where \mathcal{F}_1 is the class of linear functions and \mathcal{F}_2 is the class of quadratic functions. Then, the best function class under CV is

$$\mathcal{F}_* = \arg \min_{\mathcal{F} \in \mathbf{F}} \frac{1}{K} \sum_{k=1}^K L(\text{Alg}(\mathcal{F}, \hat{\mathcal{D}}_k), \tilde{\mathcal{D}}_k). \quad (5)$$

After the best function class is chosen, a model is trained on the entire dataset as $f_* = \text{Alg}(\mathcal{F}_*, \mathcal{D})$.

5 Off-Policy Cross-Validation

Now, we adapt cross-validation to off-policy evaluation. In supervised learning, we do not know the true data distribution but are given samples from it. Each individual sample is an unbiased but noisy estimate of the true value. Similarly, we do not know the true value of policy π in off-policy evaluation. However, we have samples collected by another policy π_0 and thus can estimate $V(\pi)$.

To formalize this observation, let $\tilde{V}(\pi; \tilde{\mathcal{D}}_k)$ be an unbiased *validator*, such as \hat{V}_{IPS} or \hat{V}_{DR} in Section 2, that estimates the true value from a *validation set* $\tilde{\mathcal{D}}_k$. Let $\hat{V}(\pi; \hat{\mathcal{D}}_k)$ be an *evaluated estimator* on a *training set* $\hat{\mathcal{D}}_k$. Then the squared loss of the evaluated estimator $\hat{V}_k = \hat{V}(\pi; \hat{\mathcal{D}}_k)$ with respect to the validator $\tilde{V}_k = \tilde{V}(\pi; \tilde{\mathcal{D}}_k)$ is

$$L(\hat{V}_k, \tilde{V}_k) = (\tilde{V}_k - \hat{V}_k)^2. \quad (6)$$

Unlike in supervised learning (Section 4), the loss is only over one observation, an unbiased estimate of $V(\pi)$. As in supervised learning, we randomly split the dataset \mathcal{D} into $\hat{\mathcal{D}}_k$ and $\tilde{\mathcal{D}}_k$, for K times. The average loss of an estimator \hat{V} on a held-out validation set is $\frac{1}{K} \sum_{k=1}^K L(\hat{V}_k, \tilde{V}_k)$. In contrast to Section 4, we use *Monte Carlo cross-validation* (Xu and Liang 2001) because we need to control the sizes of $\hat{\mathcal{D}}_k$ and $\tilde{\mathcal{D}}_k$ independently from the number of splits.

The average loss on a held-out set can be used to select an estimator as follows. Suppose that we have a set of estimators \mathbf{V} . For instance, if $\mathbf{V} = \{\hat{V}_{\text{IPS}}, \hat{V}_{\text{DM}}, \hat{V}_{\text{DR}}\}$, the set contains IPS, DM, and DR (Section 2). Then, the best estimator under CV can be defined similarly to (5) as

$$\hat{V}_* = \arg \min_{\hat{V} \in \mathbf{V}} \frac{1}{K} \sum_{k=1}^K L(\hat{V}_k, \tilde{V}_k). \quad (7)$$

After the best estimator is chosen, we return the estimated policy value from the entire dataset \mathcal{D} , $\hat{V}_*(\pi; \mathcal{D})$. This is the key idea in our proposed method.

To make the algorithm practical, we need to control the variances of the evaluated estimator and validator. The rest of Section 5 contains an analysis that provides insights into this problem. We also make (7) more robust.

Analysis

We would like to choose an estimator that minimizes the true squared loss

$$(V(\pi) - \hat{V}(\pi))^2, \quad (8)$$

where $\hat{V}(\pi) = \hat{V}(\pi; \mathcal{D})$ is its evaluated estimate on dataset \mathcal{D} and $V(\pi)$ is the true policy value. This cannot be done because $V(\pi)$ is unknown. On the other hand, if $V(\pi)$ was known, we would not have an off-policy estimation problem. In this analysis, we show that the minimized loss in (7) is a good proxy for (8).

We make the following assumptions. The only randomness in our analysis is in how \mathcal{D} is split into the training set $\hat{\mathcal{D}}_k$ and validation set $\tilde{\mathcal{D}}_k$. The sizes of these sets are \hat{n} and \tilde{n} , respectively, and $\hat{n} + \tilde{n} = n$. Let $\hat{V}_k = \hat{V}(\pi; \hat{\mathcal{D}}_k)$ be the

value of policy π estimated by the evaluated estimator on $\hat{\mathcal{D}}_k$ and $\hat{\mu} = \mathbb{E}[\hat{V}_k]$ be its mean. Let $\tilde{V}_k = \tilde{V}(\pi; \tilde{\mathcal{D}}_k)$ be the value of policy π estimated by the validator on $\tilde{\mathcal{D}}_k$ and $\tilde{\mu} = \mathbb{E}[\tilde{V}_k]$ be its mean. Using this notation, the true loss in (8) can be bounded from above as follows.

Theorem 1. For any split $k \in [K]$,

$$\begin{aligned} (\hat{V}(\pi) - V(\pi))^2 &\leq 2\mathbb{E}[(\hat{V}_k - \tilde{V}_k)^2] + \\ &\quad 4\mathbb{E}[(\hat{V}_k - \hat{\mu})^2] + 4\mathbb{E}[(\tilde{V}_k - \tilde{\mu})^2] + \\ &\quad 4(\hat{\mu} - \hat{V}(\pi))^2 + 4(\tilde{\mu} - V(\pi))^2. \end{aligned}$$

Proof. The proof uses independence assumptions and that

$$(a + b)^2 \leq 2(a^2 + b^2) \quad (9)$$

holds for any $a, b \in \mathbb{R}$. As a first step, we introduce random \hat{V}_k and \tilde{V}_k , and then apply (9),

$$\begin{aligned} (\hat{V}(\pi) - V(\pi))^2 &= \mathbb{E}[(\hat{V}(\pi) - \hat{V}_k + \hat{V}_k - V(\pi) + \tilde{V}_k - \tilde{V}_k)^2] \\ &\leq 2\mathbb{E}[(\hat{V}_k - \tilde{V}_k)^2] + 2\mathbb{E}[(\hat{V}(\pi) - \hat{V}_k - V(\pi) + \tilde{V}_k)^2]. \end{aligned}$$

Using (9) again, we bound the last term from above by

$$4\mathbb{E}[(\hat{V}_k - \hat{V}(\pi))^2] + 4\mathbb{E}[(\tilde{V}_k - V(\pi))^2].$$

Since $\hat{\mu} = \mathbb{E}[\hat{V}_k]$ and $\hat{\mu} - \hat{V}(\pi)$ is fixed, we get

$$\begin{aligned} \mathbb{E}[(\hat{V}_k - \hat{V}(\pi))^2] &= \mathbb{E}[(\hat{V}_k - \hat{\mu} + \hat{\mu} - \hat{V}(\pi))^2] \\ &= \mathbb{E}[(\hat{V}_k - \hat{\mu})^2] + (\hat{\mu} - \hat{V}(\pi))^2. \end{aligned}$$

Similarly, since $\tilde{\mu} = \mathbb{E}[\tilde{V}_k]$ and $\tilde{\mu} - V(\pi)$ is fixed, we get

$$\mathbb{E}[(\tilde{V}_k - V(\pi))^2] = \mathbb{E}[(\tilde{V}_k - \tilde{\mu})^2] + (\tilde{\mu} - V(\pi))^2.$$

Finally, we chain all inequalities and get our claim. \square

The bound in Theorem 1 can be viewed as follows. The first term $\mathbb{E}[(\hat{V}_k - \tilde{V}_k)^2]$ is the expectation of our optimized loss (Theorem 2). The second term is the variance of the evaluated estimator on a training set of size \hat{n} , and thus is $\mathcal{O}(\hat{\sigma}^2/\hat{n})$ for some $\hat{\sigma}^2 > 0$. The third term is the variance of the validator on a validation set of size \tilde{n} , and thus is $\mathcal{O}(\tilde{\sigma}^2/\tilde{n})$ for some $\tilde{\sigma}^2 > 0$. The fourth term is zero for any unbiased off-policy estimator in Section 2. We assume that $\hat{\mu} = \hat{V}(\pi)$ in our discussion. Finally, the last term is the difference between the unbiased estimate of the value of policy π on \mathcal{D} and $V(\pi)$. This term is $\mathcal{O}(\log(1/\delta)/n)$ with probability at least $1 - \delta$ by standard concentration inequalities (Boucheron, Lugosi, and Massart 2013), since \mathcal{D} is a sample of size n . Based on our discussion,

$$\begin{aligned} (\hat{V}(\pi) - V(\pi))^2 &\leq 2\mathbb{E}[(\hat{V}_k - \tilde{V}_k)^2] + \\ &\quad \mathcal{O}(\hat{\sigma}^2/\hat{n} + \tilde{\sigma}^2/\tilde{n}) + \mathcal{O}(\log(1/\delta)/n) \end{aligned}$$

holds with probability at least $1 - \delta$.

The above bound can be minimized as follows. The last term measures how representative the dataset \mathcal{D} is. This is out of our control. To minimize $\mathcal{O}(\hat{\sigma}^2/\hat{n} + \tilde{\sigma}^2/\tilde{n})$, we set \hat{n} and

\tilde{n} proportionally to the variances of the evaluated estimator and validator,

$$\hat{n} = \frac{\hat{\sigma}^2}{\hat{\sigma}^2 + \tilde{\sigma}^2}n, \quad \tilde{n} = \frac{\tilde{\sigma}^2}{\hat{\sigma}^2 + \tilde{\sigma}^2}n,$$

respectively. Finally, we relate $\mathbb{E}[(\hat{V}_k - \tilde{V}_k)^2]$ to our minimized loss in (7).

Theorem 2. For any split $\ell \in [K]$,

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^K(\hat{V}_k - \tilde{V}_k)^2\right] = \mathbb{E}[(\hat{V}_\ell - \tilde{V}_\ell)^2].$$

The variance of the estimator is

$$\text{var}\left[\frac{1}{K}\sum_{k=1}^K(\hat{V}_k - \tilde{V}_k)^2\right] = \mathcal{O}(1/K).$$

Proof. The first claim follows from the linearity of expectation and that $\hat{V}_k - \tilde{V}_k$ are drawn independently from the same distribution. To prove the second claim, we rewrite the variance of the estimator as

$$\frac{\sum_{i,j=1}^K \mathbb{E}[(\hat{V}_i - \tilde{V}_i)^2(\hat{V}_j - \tilde{V}_j)^2]}{K^2} - \mathbb{E}[(\hat{V}_k - \tilde{V}_k)^2]^2 \quad (10)$$

using $\text{var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$. Because the random splits are independent,

$$\mathbb{E}[(\hat{V}_i - \tilde{V}_i)^2(\hat{V}_j - \tilde{V}_j)^2] = \mathbb{E}[(\hat{V}_k - \tilde{V}_k)^2]^2$$

for any $i \neq j$. This happens exactly $K(K-1)$ times out of K^2 . As a result, (10) can be rewritten as

$$\frac{1}{K}\mathbb{E}[(\hat{V}_k - \tilde{V}_k)^4] - \frac{1}{K}\mathbb{E}[(\hat{V}_k - \tilde{V}_k)^2]^2 = \mathcal{O}(1/K).$$

This concludes the proof. \square

Theorem 2 says that the estimated loss from K random splits concentrates at $\mathbb{E}[(\hat{V}_k - \tilde{V}_k)^2]$ at rate $\mathcal{O}(1/\sqrt{K})$. Hence, by standard concentration inequalities (Boucheron, Lugosi, and Massart 2013),

$$\begin{aligned} (\hat{V}(\pi) - V(\pi))^2 &\leq \frac{2}{K}\sum_{k=1}^K(\hat{V}_k - \tilde{V}_k)^2 + \\ &\quad \mathcal{O}(\hat{\sigma}^2/\hat{n} + \tilde{\sigma}^2/\tilde{n}) + \\ &\quad \mathcal{O}(\log(1/\delta)/n) + \mathcal{O}(\sqrt{\log(1/\delta'}/K)) \end{aligned}$$

holds with probability at least $1 - \delta - \delta'$. The last term can be driven to zero with more random splits K .

One Standard Error Rule

If the set of estimators \mathbf{V} in (7) is large, we could choose a poor estimator that performs well just by chance with a high probability. This problem is exacerbated in small datasets (Varma and Simon 2006). To account for this in supervised CV, Hastie, Tibshirani, and Friedman (2009) proposed a heuristic called the *one standard error rule*. This heuristic chooses the simplest model whose performance is within one

Algorithm 1: Off-policy evaluation with cross-validated estimator selection.

```

1: Input: Evaluated policy  $\pi$ , logged dataset  $\mathcal{D}$ , set of estimators  $\mathbf{V}$ , number of random splits  $K$ 
2:  $\hat{\sigma}^2 \leftarrow$  Empirical estimate of  $\text{var} [\hat{V}(\pi; \mathcal{D})]$ 
3: for  $\hat{V} \in \mathbf{V}$  do
4:    $\hat{\sigma}^2 \leftarrow$  Empirical estimate of  $\text{var} [\hat{V}(\pi; \mathcal{D})]$ 
5:   for  $k = 1, \dots, K$  do
6:      $\hat{\mathcal{D}}_k, \tilde{\mathcal{D}}_k \leftarrow$  Split  $\mathcal{D}$  such that  $|\hat{\mathcal{D}}_k|/|\tilde{\mathcal{D}}_k| = \hat{\sigma}^2/\tilde{\sigma}^2$ 
7:      $L_{\hat{V},k} \leftarrow (\hat{V}(\pi; \hat{\mathcal{D}}_k) - \hat{V}(\pi; \tilde{\mathcal{D}}_k))^2$ 
8:   end for
9:    $\bar{L}_{\hat{V}} \leftarrow \frac{1}{K} \sum_{k=1}^K L_{\hat{V},k}$ 
10: end for
11:  $\hat{V}_* \leftarrow \arg \min_{\hat{V} \in \mathbf{V}} \bar{L}_{\hat{V}} + \sqrt{\frac{1}{K-1} \sum_{k=1}^K (L_{\hat{V},k} - \bar{L}_{\hat{V}})^2}$ 
12: Output:  $\hat{V}_*(\pi; \mathcal{D})$ 

```

standard error of the best model. Roughly speaking, these models cannot be statistically distinguished.

Inspired by the one standard error rule, we choose an estimator with the *lowest one-standard-error upper bound* on its loss. This is also known as *pessimistic optimization* (Buckman, Gelada, and Bellemare 2020). Compared to the original rule (Hastie, Tibshirani, and Friedman 2009), we do not need to know which estimator has the lowest complexity.

Algorithm

We call our method **Off-policy Cross-Validation (OCV)** and present its pseudo-code in Algorithm 1. The method works as follows. First, we estimate the variance of the validator \hat{V} (line 2). Details are provided in Appendix A. Second, we estimate the variance of each evaluated estimator \hat{V} (line 4). Third, we repeatedly split \mathcal{D} into the training and validation sets (line 6) and calculate the loss of the evaluated estimator with respect to the validator (line 7). Finally, we select the estimator \hat{V}_* with the lowest one-standard-error upper bound on its estimated loss (line 11).

6 Experiments

We conduct three main experiments. First, we evaluate OCV on an estimator selection problem among IPS, DM, and DR. Second, we apply OCV to hyper-parameter tuning of seven other estimators. We compare against SLOPE, PAS-IF, and estimator-specific tuning heuristics if the authors provided one. Finally, we show that OCV can jointly choose the best estimator and its hyper-parameters, and thus is a practical method to get a high-quality estimator. Appendix B contains ablation studies on the individual components of OCV and computational efficiency. We also show the importance of having an unbiased validator and that OCV performs well even in low-data regimes.

Datasets. We take nine UCI datasets (Markelle, Longjohn,

and Nottingham 2023) and convert them into contextual bandit problems, similarly to prior works (Dudik et al. 2014; Wang, Agarwal, and Dudik 2017; Farajtabar, Chow, and Ghavamzadeh 2018; Su et al. 2019, 2020). The datasets have different characteristics (Appendix A), such as sample size and the number of features, and thus cover a wide range of potential applications of our method. Each dataset contains n examples, $\mathcal{H} = \{(x_i, y_i)\}_{i \in [n]}$, where $x_i \in \mathbb{R}^d$ and $y_i \in [m]$ are the feature vector and label of example i , respectively; and m is the number of classes. We split each \mathcal{H} into two halves, the bandit feedback dataset \mathcal{H}_b and policy learning dataset \mathcal{H}_π .

The *bandit feedback dataset* is used to compute the policy value and log data. Specifically, the value of policy π is

$$V(\pi) = \frac{1}{|\mathcal{H}_b|} \sum_{(x,y) \in \mathcal{H}_b} \sum_{a=1}^m \pi(a|x) \mathbb{1}\{a=y\}.$$

The logged dataset \mathcal{D} has the same size as \mathcal{H}_b , $n = |\mathcal{H}_b|$, and is defined as

$$\mathcal{D} = \{(x, a, \mathbb{1}\{a=y\}) : a \sim \pi_0(\cdot|x), (x, y) \in \mathcal{H}_b\}.$$

For each example in \mathcal{H}_b , the logging policy π_0 takes an action conditioned on its feature vector. The reward is one if the index of the action matches the label and zero otherwise.

The *policy learning dataset* is used to estimate π and π_0 . We proceed as follows. First, we take a bootstrap sample of \mathcal{H}_π of size $|\mathcal{H}_\pi|$ and learn a logistic model for each class $a \in [m]$. Let $\theta_{a,0} \in \mathbb{R}^d$ be the learned logistic model parameter for class a . Second, we take another bootstrap sample of \mathcal{H}_π of size $|\mathcal{H}_\pi|$ and learn a logistic model for each class $a \in [m]$. Let $\theta_{a,1} \in \mathbb{R}^d$ be the learned logistic model parameter for class a in the second bootstrap sample. Based on $\theta_{a,0}$ and $\theta_{a,1}$, we define our policies as

$$\begin{aligned} \pi_0(a|x) &= \frac{\exp(\beta_0 x^\top \theta_{a,0})}{\sum_{a'=1}^m \exp(\beta_0 x^\top \theta_{a',0})}, \\ \pi(a|x) &= \frac{\exp(\beta_1 x^\top \theta_{a,1})}{\sum_{a'=1}^m \exp(\beta_1 x^\top \theta_{a',1})}. \end{aligned} \quad (11)$$

The parameters β_0 and β_1 are *inverse temperatures* of the softmax function. Positive values prefer high-value actions and vice versa. The zero temperature is a uniform policy. The temperatures β_0 and β_1 are chosen later in each experiment. We take two bootstrap samples to ensure that π and π_0 are not simple transformations of each other.

Our method and baselines. We evaluate two variants of our method, OCV_{IPS} and OCV_{DR} , with IPS and DR as validators. OCV is implemented as described in Algorithm 1 with $K = 10$. The reward model \hat{f} in all relevant estimators is learned using ridge regression with a regularization coefficient 0.001. We consider two baselines: SLOPE and PAS-IF (Section 3). In the tuning experiment (Section 6), we also implement the original tuning procedures if the authors provided one. All implementation details are in Appendix A.

Estimator Selection

We want to choose the best estimator from three candidates: IPS in (1), DM in (2), and DR in (3). We use $\beta_0 = 1$ for

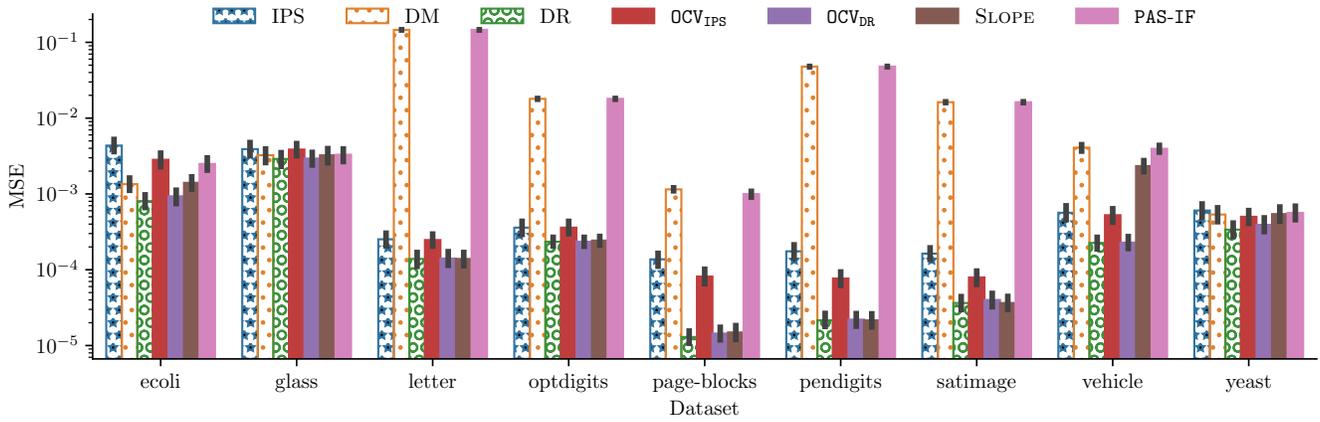


Figure 1: MSE of our estimator selection methods, OCV_{IPS} and OCV_{DR} , compared against two other estimator selection baselines, SLOPE and PAS-IF. The methods select the best estimator out of IPS, DM, and DR. In all figures, we report 95% confidence intervals estimated by bootstrapping.

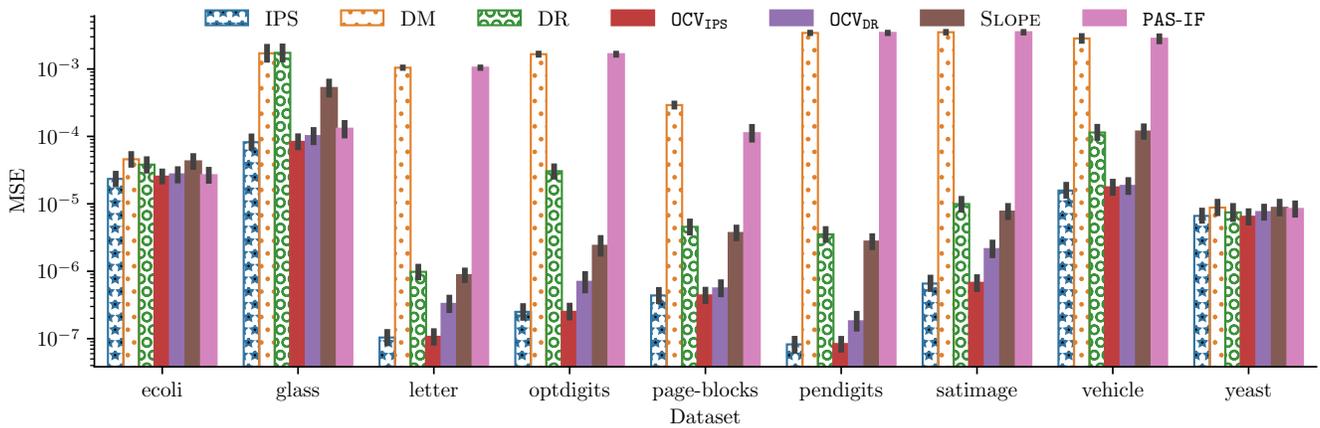


Figure 2: MSE of the methods for temperatures $\beta_0 = 1$ and $\beta_1 = -10$. OCV performs well even when its validator does not, for example OCV_{DR} on the *glass* dataset. This also shows that OCV does not simply choose the same estimator as the validator.

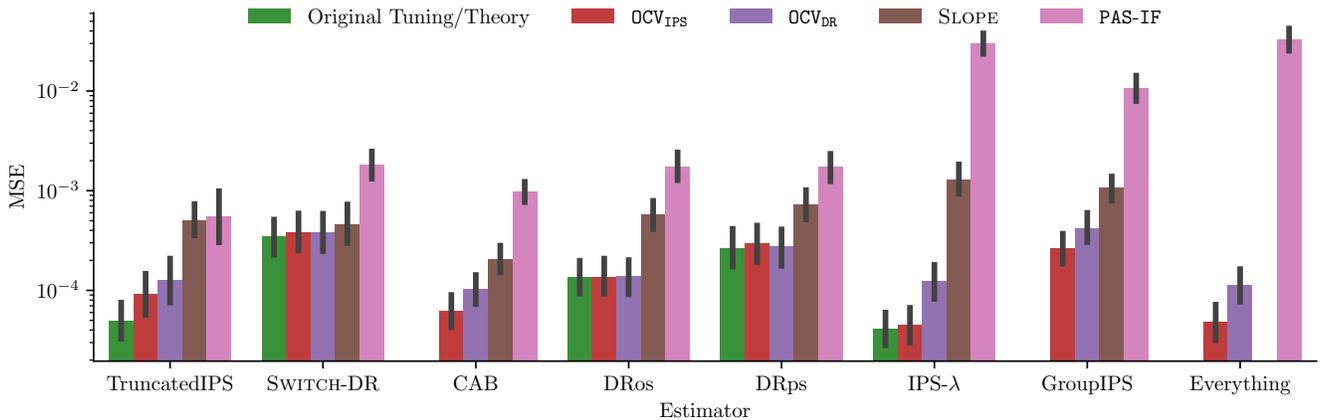


Figure 3: MSE of our estimator selection methods and specialized theoretical approaches applied to hyper-parameter tuning of various estimators. *Everything* refers to the joint estimator selection and hyper-parameter tuning. This shows that OCV is a reliable and practical method for choosing a suitable and well-tuned estimator.

the logging policy and $\beta_1 = 10$ for the target policy. This is a realistic scenario where the logging policy prefers high-value actions, and the target policy takes them even more often. SLOPE requires the estimators to be ordered by their variances. This may not be always possible. However, the bias-variance trade-offs of IPS, DM, and DR are generally

$$\text{var} \left[\hat{V}_{\text{IPS}}(\pi) \right] \geq \text{var} \left[\hat{V}_{\text{DR}}(\pi) \right] \geq \text{var} \left[\hat{V}_{\text{DM}}(\pi) \right]$$

and we use this order. All our results are averaged over 500 independent runs. A new run always starts by splitting the dataset into the bandit feedback and policy learning datasets, as described earlier.

Cross-validation consistently chooses a good estimator.

Figure 1 shows that our methods avoid the worst estimator and perform better on average than both SLOPE and PAS-IF. OCV_{DR} significantly outperforms all methods on two datasets while never being much worse. We observe that SLOPE performs well because its bias-variance assumptions are satisfied. PAS-IF prefers DM even though it performs poorly. We hypothesize that this is because the tuning procedure of PAS-IF is biased. As we show in Appendix B, a biased validator tends to prefer similarly biased estimators and thus cannot be reliably used for estimator selection.

Cross-validation with DR performs well even when DR performs poorly.

One may think that OCV_{DR} performs well in Figure 1 because the best estimator is DR (Dudik et al. 2014). To disprove this, we change the temperature of the target policy to $\beta_1 = -10$ and show new results in Figure 2. The DR is no longer the best estimator, yet OCV_{DR} performs well. Both of our methods outperform SLOPE and PAS-IF again. We also observe that SLOPE performs worse in this experiment. Since both IPS and DR are unbiased, their confidence intervals often overlap. Therefore, SLOPE mostly chooses DR regardless of its performance.

Hyper-Parameter Tuning

We also evaluate OCV on the hyper-parameter tuning of seven estimators from Section 3. We present them next. TruncatedIPS (Ionides 2008) is parameterized by a clipping constant M that clips higher propensity weights than M . The authors suggest $M = \sqrt{n}$. SWITCH-DR (Wang, Agarwal, and Dudik 2017) has a threshold parameter τ that switches to DM if the propensity weights are too high and uses DR otherwise. The authors propose their own tuning strategy by pessimistically bounding the estimator’s bias (Wang, Agarwal, and Dudik 2017). CAB (Su et al. 2019) has a parameter M that adaptively blends DM and IPS. The authors do not propose any tuning method. DRos and DRps (Su et al. 2020) have a parameter λ that regularizes propensity weights to decrease DR’s variance. The authors propose a tuning strategy similar to that of SWITCH-DR. IPS- λ (Metelli, Russo, and Restelli 2021) has a parameter λ that regularizes propensity weights while keeping the estimates differentiable, which is useful for off-policy learning. The authors propose a differentiable tuning objective to get optimal λ . GroupIPS (Peng et al. 2023) has multiple tuning parameters, such as the number of clusters

M , the reward model class to identify similar actions, and the clustering algorithm. The authors propose choosing M by SLOPE. We describe the estimators, their original tuning procedures, and hyper-parameter grids in Appendix A.

All methods are evaluated in 90 different conditions: 9 UCI ML Repository datasets (Markelle, Longjohn, and Nottingham 2023), two target policies $\beta_1 \in \{-10, 10\}$, and five logging policies $\beta_0 \in \{-3, -1, 0, 1, 3\}$. This covers a wide range of scenarios: logging and target policies can be close or differ, their values can be high or low, and dataset sizes vary from small (107) to larger (10 000). Each condition is repeated 5 times, and we report the MSE over all runs and conditions in Figure 3. We observe that theory-suggested hyper-parameter values generally perform the best if they exist. Surprisingly, OCV often matches their performance while also being a general solution that applies to any estimator. It typically outperforms SLOPE and PAS-IF.

We also consider the problem of joint estimator selection and hyper-parameter tuning. We evaluate OCV_{DR} , OCV_{IPS} , and PAS-IF on this task and report our results as *Everything* in Figure 3. SLOPE cannot be evaluated because the correct order of the estimators is unclear. We observe that both of our estimators perform well and have an order of magnitude lower MSE than PAS-IF. This shows that OCV is a reliable and practical method.

7 Conclusion

We propose an estimator selection and hyper-parameter tuning procedure for off-policy evaluation that uses cross-validation, bridging an important gap between off-policy evaluation and supervised learning. Estimator selection in off-policy evaluation has been mostly theory-driven. In contrast, in supervised learning, cross-validation is preferred despite limited theoretical support. We overcome the issue of an unknown policy value by using an unbiased estimator on a held-out validation set. This is similar to cross-validation in supervised learning, where we only have samples from an unknown distribution. We test our method extensively on nine real-world datasets, as well as both estimator selection and hyper-parameter tuning tasks. The method is widely applicable, simple to implement, and easy to understand since it relies on principal techniques from supervised learning that are well-known and loved by practitioners. It also outperforms state-of-the-art methods.

One natural future direction is off-policy learning. The main challenge is that the tuned hyper-parameters have to work well for any policy instead of a single target policy. At the same time, naive tuning of some worst-case empirical risk could lead to too conservative choices. Another potential direction is an extension to reinforcement learning.

Acknowledgements

This research was partially supported by DisAi, a project funded by the European Union under the Horizon Europe, GA No. 101079164, <https://doi.org/10.3030/101079164>; HERMES - a project by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I03-03-V04-00336; and OZ BrAIIn association.

References

- Akaike, H. 1998. Information Theory and an Extension of the Maximum Likelihood Principle. In Parzen, E.; Tanabe, K.; and Kitagawa, G., eds., *Selected Papers of Hirotugu Akaike*, 199–213. New York, NY: Springer New York. ISBN 978-1-4612-7248-9 978-1-4612-1694-0. Series Title: Springer Series in Statistics.
- Aouali, I.; Brunel, V.-E.; Rohde, D.; and Korba, A. 2023. Exponential Smoothing for Off-Policy Learning. In *Proceedings of the 40th International Conference on Machine Learning*, 984–1017. PMLR. ISSN: 2640-3498.
- Bishop, C. M. 2006. *Pattern recognition and machine learning*. Information science and statistics. New York: Springer. ISBN 978-0-387-31073-2.
- Boucheron, S.; Lugosi, G.; and Massart, P. 2013. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press. ISBN 978-0-19-953525-5.
- Buckman, J.; Gelada, C.; and Bellemare, M. G. 2020. The Importance of Pessimism in Fixed-Dataset Policy Optimization. In *International Conference on Learning Representations*.
- Cief, M.; Golebiowski, J.; Schmidt, P.; Abedjan, Z.; and Bekasov, A. 2024. Learning Action Embeddings for Off-Policy Evaluation. In Goharian, N.; Tonello, N.; He, Y.; Lipani, A.; McDonald, G.; Macdonald, C.; and Ounis, I., eds., *Advances in Information Retrieval*, 108–122. Cham: Springer Nature Switzerland. ISBN 978-3-031-56027-9.
- Devroye, L.; Györfi, L.; and Lugosi, G. 1996. *A Probabilistic Theory of Pattern Recognition*, volume 31 of *Stochastic Modelling and Applied Probability*. New York, NY: Springer New York. ISBN 978-1-4612-6877-2 978-1-4612-0711-5.
- Dudik, M.; Erhan, D.; Langford, J.; and Li, L. 2014. Doubly Robust Policy Evaluation and Optimization. *Statistical Science*, 29(4): 485–511.
- Farajtabar, M.; Chow, Y.; and Ghavamzadeh, M. 2018. More Robust Doubly Robust Off-policy Evaluation. In *Proceedings of the 35th International Conference on Machine Learning*, 1447–1456. PMLR. ISSN: 2640-3498.
- Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY: Springer New York. ISBN 978-0-387-84857-0 978-0-387-84858-7.
- Hauskrecht, M.; and Fraser, H. 2000. Planning treatment of ischemic heart disease with partially observable Markov decision processes. *Artificial Intelligence in Medicine*, 18(3): 221–244.
- Ionides, E. L. 2008. Truncated Importance Sampling. *Journal of Computational and Graphical Statistics*, 17(2): 295–311.
- Kallus, N.; and Zhou, A. 2018. Policy Evaluation and Optimization with Continuous Treatments. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, 1243–1251. PMLR. ISSN: 2640-3498.
- Langford, J.; Strehl, A.; and Wortman, J. 2008. Exploration scavenging. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, 528–535. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-60558-205-4.
- Lee, J.; Tucker, G.; Nachum, O.; and Dai, B. 2022. Model Selection in Batch Policy Optimization. In *Proceedings of the 39th International Conference on Machine Learning*, 12542–12569. PMLR. ISSN: 2640-3498.
- Lepski, O. V.; and Spokoiny, V. G. 1997. Optimal pointwise adaptive methods in nonparametric estimation. *The Annals of Statistics*, 25(6): 2512–2546. Publisher: Institute of Mathematical Statistics.
- Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, WWW '10, 661–670. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-60558-799-8.
- Markelle, K.; Longjohn, R.; and Nottingham, K. 2023. UCI Machine Learning Repository.
- Metelli, A. M.; Russo, A.; and Restelli, M. 2021. Subgaussian and Differentiable Importance Sampling for Off-Policy Evaluation and Learning. In *Advances in Neural Information Processing Systems*, volume 34, 8119–8132. Curran Associates, Inc.
- Nie, A.; Flet-Berliac, Y.; Jordan, D.; Steenbergen, W.; and Brunskill, E. 2022. Data-Efficient Pipeline for Offline Reinforcement Learning with Limited Data. *Advances in Neural Information Processing Systems*, 35: 14810–14823.
- Peng, J.; Zou, H.; Liu, J.; Li, S.; Jiang, Y.; Pei, J.; and Cui, P. 2023. Offline Policy Evaluation in Large Action Spaces via Outcome-Oriented Action Grouping. In *Proceedings of the ACM Web Conference 2023*, WWW '23, 1220–1230. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-9416-1.
- Robins, J. M.; Rotnitzky, A.; and Zhao, L. P. 1994. Estimation of Regression Coefficients When Some Regressors are not Always Observed. *Journal of the American Statistical Association*, 89(427): 846–866. Publisher: Taylor & Francis .eprint: <https://doi.org/10.1080/01621459.1994.10476818>.
- Sachdeva, N.; Wang, L.; Liang, D.; Kallus, N.; and McAuley, J. 2024. Off-Policy Evaluation for Large Action Spaces via Policy Convolution. In *Proceedings of the ACM on Web Conference 2024*, WWW '24, 3576–3585. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701719.
- Saito, Y.; and Joachims, T. 2022. Off-Policy Evaluation for Large Action Spaces via Embeddings. In *Proceedings of the 39th International Conference on Machine Learning*, 19089–19122. PMLR. ISSN: 2640-3498.
- Saito, Y.; and Nomura, M. 2024. Hyperparameter Optimization Can Even be Harmful in Off-Policy Learning and How to Deal with It. ArXiv:2404.15084 [cs].
- Saito, Y.; Ren, Q.; and Joachims, T. 2023. Off-Policy Evaluation for Large Action Spaces via Conjunct Effect Modeling. In *Proceedings of the 40th International Conference on Machine Learning*, 29734–29759. PMLR. ISSN: 2640-3498.
- Saito, Y.; Udagawa, T.; Kiyohara, H.; Mogi, K.; Narita, Y.; and Tateno, K. 2021. Evaluating the Robustness of Off-Policy Evaluation. In *Proceedings of the 15th ACM Conference on*

- Recommender Systems*, RecSys '21, 114–123. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-8458-2.
- Schwarz, G. 1978. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2).
- Silver, D.; Newnham, L.; Barker, D.; Weller, S.; and McFall, J. 2013. Concurrent Reinforcement Learning from Customer Interactions. In *Proceedings of the 30th International Conference on Machine Learning*, 924–932. PMLR. ISSN: 1938-7228.
- Stoica, P.; and Selen, Y. 2004. Model-order selection. *IEEE Signal Processing Magazine*, 21(4): 36–47.
- Stone, M. 1974. Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 36(2): 111–133.
- Su, Y.; Dimakopoulou, M.; Krishnamurthy, A.; and Dudik, M. 2020. Doubly robust off-policy evaluation with shrinkage. In *Proceedings of the 37th International Conference on Machine Learning*, 9167–9176. PMLR. ISSN: 2640-3498.
- Su, Y.; Srinath, P.; and Krishnamurthy, A. 2020. Adaptive Estimator Selection for Off-Policy Evaluation. In *Proceedings of the 37th International Conference on Machine Learning*, 9196–9205. PMLR. ISSN: 2640-3498.
- Su, Y.; Wang, L.; Santacatterina, M.; and Joachims, T. 2019. CAB: Continuous Adaptive Blending for Policy Evaluation and Learning. In *Proceedings of the 36th International Conference on Machine Learning*, 6005–6014. PMLR. ISSN: 2640-3498.
- Udagawa, T.; Kiyohara, H.; Narita, Y.; Saito, Y.; and Tateno, K. 2023. Policy-Adaptive Estimator Selection for Off-Policy Evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 10025–10033.
- Varma, S.; and Simon, R. 2006. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7(1): 91.
- Vlassis, N.; Bibaut, A.; Dimakopoulou, M.; and Jebara, T. 2019. On the Design of Estimators for Bandit Off-Policy Evaluation. In *Proceedings of the 36th International Conference on Machine Learning*, 6468–6476. PMLR. ISSN: 2640-3498.
- Voloshin, C.; Le, H.; Jiang, N.; and Yue, Y. 2021. Empirical Study of Off-Policy Policy Evaluation for Reinforcement Learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.
- Wang, Y.-X.; Agarwal, A.; and Dudik, M. 2017. Optimal and Adaptive Off-policy Evaluation in Contextual Bandits. In *Proceedings of the 34th International Conference on Machine Learning*, 3589–3597. PMLR. ISSN: 2640-3498.
- Xu, Q.-S.; and Liang, Y.-Z. 2001. Monte Carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56(1): 1–11.
- Yao, Y.; Rosasco, L.; and Caponnetto, A. 2007. On Early Stopping in Gradient Descent Learning. *Constructive Approximation*, 26(2): 289–315.