Local Metric Learning for Off-Policy Evaluation in Contextual Bandits with Continuous Actions

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider local kernel metric learning for off-policy policy evaluation (OPE) of deterministic policies in contextual bandits with continuous action spaces. Our work is motivated by practical scenarios where the target policy needs to be deterministic due to domain requirements, such as prescription of treatment dosage and duration in medicine. Although importance sampling (IS) provides a basic principle for OPE, it is ill-posed for the deterministic target policy with continuous actions. Our main idea is to relax the target policy and pose the problem as kernel-based estimation, while learning the kernel metric in order to minimize the overall mean square error (MSE). We present an analytic solution for the optimal metric, based on the analysis of bias and variance. Whereas prior work has been limited to scalar action spaces or kernel bandwidth optimization, our work takes a step further being capable of vector action spaces and metric optimization. We show that our estimator is consistent, and significantly reduces MSE compared to baseline OPE methods through experiments on various domains.

1 Introduction

In order to deploy a contextual bandit policy to an actual environment, such as personalized pricing[1], treatments [2], recommendation [3], advertisements [4], the performance of the policy being deployed should be evaluated to decide whether the trained policy is suitable for the deployment. This is because the interaction of the policy with the environment could be costly and dangerous. For example, using an erroneous medical treatment policy to prescribe drugs for patients could result in dire consequences. Therefore, emerges a necessity for an algorithm that can evaluate a policy's performance without having it interact with the environment in an on-line manner. Such algorithm is called the "off-policy policy evaluation" (OPE) algorithms. [5]. OPE algorithms for contextual bandits evaluate a target policy by estimating the expected reward the it would get given an offline data sampled from a data collecting (behavior) policy.

Previous work on OPE for contextual bandits has mainly focused on environments with finite actions [4–10]. The work can be largely divided into three approaches [6, 9, 11]. The first approach is direct method (DM) which learns the environment model to evaluate a target policy. DM learns to model the environment for estimating the expected reward, which is used for policy evaluation. Since the environment model is learned with a function approximation, the estimator is biased. Although it is biased, it is known to have a low variance [9]. The second approach is importance sampling (IS) which corrects the data distribution induced by a behavior policy to that of a target policy [12], and uses the corrected distribution to approximate the expected reward of the target policy. IS estimate is unbiased when behavior and target policies are given. However, IS estimates can have a large variance when there is large mismatch between behavior and target policy distributions [9]. The last approach is doubly robust (DR), which uses DM to reduce the variance of IS while keeping the unbiasedness of an IS estimate [6].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

However, there is also a need for OPE algorithms for contextual bandits with continuous actions, as it can be used to evaluate the policies for personalized continuous decision making, such as evaluating a continuous treatment prescription policies. [2, 13, 14]. And due to practical constraints, we also need OPE algorithms for deterministic contextual bandit policies. For example, treatment prescription policies cannot stochastically prescribe drugs to patients. Unfortunately, most studies for OPE with finite actions cannot be easily extended to evaluate deterministic contextual bandit policies with continuous actions. This is because IS weights used for both IS and DR estimates are almost surely zero for a deterministic target policies in continuous action space. Thus, the work for evaluating deterministic contextual bandit policies continuous action space focus on this problem [2, 14, 15].

The work on OPE of deterministic contextual bandits with continuous actions can be largely divided into work on IS [2, 15] and doubly robust estimators (DR) [13, 14]. Both methods use IS and eliminate the problem of almost surely having zero IS estimate when given a deterministic target policy and a stochastic behavior policy in continuous action space. Some work relaxes the deterministic policy to a kernel [2, 13, 15], and the other work discretizes the action space [14]. These methods reduce the mismatch between behavior and target policies by relaxing the Dirac-delta target policies (a deterministic target policy density can be seen as a Dirac-delta function [2]) to a kernel function and thus reduce the variance with some addition of bias.

The kernel-based IS methods for OPE use kernels to measure the similarities between the target and behavior actions. Furthermore, the similarities are measured according to the Euclidean distances between the actions [2, 13, 15]. However, the Euclidean distance is inadequate for measuring the similarities in the case of kernel-based IS for OPE. In the OPE case, assigning high similarity (or, assigning high importance weight) to the actions having similar rewards will induce less bias for the OPE estimate when relaxing the Dirac-delta target policy density to a kernel. In kernel learning literature, there are studies that reduce the bias of kernel regression [16] and nearest neighbor classification [17] by learning kernel metrics. Similar to these methods, we focus on learning kernel metric that reduces estimation bias (definition in Section 3, Eq. (7)) of kernel-based IS methods.

We propose local kernel metric learning for IS (KMIS) for OPE estimation of deterministic contextual bandit policies with multiple continuous actions. Our proposed method learns a Mahalanobis distance metric [16–19] locally at each state that shortens the distance between behavior and target actions having similar rewards. The metric applied kernels measure similarities between actions according to the Mahalanobis distances induced by the metric. In our work, we first analytically show that the bias of a kernel-based IS estimator is a dominant factor given the optimal bandwidth that minimizes the leading MSE without a metric. Then we derive the kernel metric that minimizes the leading order bias given the optimal bandwidth. In the experiments section, we show that when the proposed kernel metric learning is used with the previous works in kernel-based IS that learn bandwidths, MSEs of the estimations are significantly reduced. We report empirical results in various synthetic domains and in real-world data.

2 Preliminaries

2.1 Problem Setting

In this work, we focus on off-policy policy evaluation (OPE) of a deterministic target policy in an environment with the action space $\mathcal{A} \subset \mathbb{R}^D$, state space $\mathcal{S} \subset \mathbb{R}^{D_S}$, the distribution of the reward $p(r|s,a)$, the state distribution $p(s)$. The target policy which selects an action given a state $\pi(s) : \mathcal{S} \to \mathcal{A}$ can be regarded to have a Dirac-delta distribution $\pi(a|s) = \delta(\pi(s) - a)$, where the probability density value goes to infinity at the selected action given a state, and the integration of the function is one. The offline data $D = \{s_i, a_i, r_i\}_{i=1}^N$ used for the evaluation of the deterministic target policy π is sampled with a known stochastic behavior policy $\pi_b(a|s) : \mathcal{S} \to \Delta(\mathcal{A})$. We assume that the support of the behavior policy $\pi_b(a|s)$ contains the actions selected by the target policy $\pi(s)$.

One of the methods to evaluate the target policy value $\rho^\pi = \mathbb{E}_{s\sim(s),a\sim\pi(a|s),r\sim p(r|s,a)}[r]$ with the offline data D sampled from π_b is to do importance sampling (IS) estimation as in Eq. (1). In our case, the IS ratios correct the action sampling distribution $\pi_b(a|s)$ to $\pi(a|s)$. But since $\pi(a|s)$ is a Dirac-delta function, the probability density function (PDF) value at the action in the offline data D which is sampled with $\pi_b(a|s)$ is almost surely 0. Existing works deal with the problem by relaxing the target policy π, which is a Dirac-delta function, to an isotropic kernel K and enable IS estimation of $\hat{\rho}^K$ as in Eq. (2) [2, 13, 15].

$$\rho^{\pi} = \mathbb{E}_{s\sim p(s), a\sim\pi_b(a|s), r\sim p(r|s,a)}\left[\frac{\pi(a \mid s)}{\pi_b(a \mid s)} r\right], \tag{1}$$

$$\left(\text{where, } \pi(a|s) = \begin{cases} \infty, & \pi(s) = a \\ 0, & \pi(s) \neq a \end{cases}, \int_{-\infty}^{\infty} \pi(a|s) ds = 1\right)$$

$$\approx \frac{1}{Nh^D} \sum_{i=1}^{N} K\left(\frac{\pi(s_i) - a_i}{h}\right) \frac{r_i}{\pi_b(a_i \mid s_i)} \tag{2}$$

2.2 Bandwidth Learning for the Isotropic Kernel Relaxed IS Estimator

As the Dirac-delta function can be regarded as a kernel having its bandwidth approaching zero, the relaxation of the Dirac-delta function to a kernel can be seen as increasing its bandwidth h. By the relaxation, the bias (definition in Section 3, Eq. (7)) of the kernel-relaxed IS estimate increases while its variance is reduced. However, as the bias and the variance compose the MSE of the estimate, the bandwidth that best balances between them should be selected. And previous works focus on choosing the bandwidth of the relaxed kernel [2, 15]. In the works of Su et al. [15], they proposed a bandwidth selection method that selects a bandwidth among the list of given bandwidths by using Lepski's principle for bandwidth selection in nonparametric statistics [20–22]. Kallus and Zhou [2] derived the leading order MSE (LOMSE) in terms of bandwidth h, sample size D, and action dimension D which is denoted as $\text{LOMSE}(h, N, D)$ in Eq. (3).

$$\text{LOMSE}(h, N, D) = h^4 C_b + \frac{C_v}{Nh^D} \tag{3}$$

$$C_b := \left(\frac{1}{2}\mathbb{E}_{s\sim p(s)} \nabla_a^2 \left(\mathbb{E}[r|s,a]\right) |_{a=\pi(\mathbf{s})}\right)^2, C_v := R(K)\mathbb{E}_{s\sim p(s)}\left[\frac{\mathbb{E}[r^2|s, a=\pi(s)]}{\pi_b(a=\pi(s)|s)}\right],$$

where, C_b and C_v are constants related to the bias (definition in Eq. (7)) and the variance of the estimation, respectively, $\nabla_{\mathbf{a}}^2$ denotes the Laplacian operator w.r.t. action $\mathbf{a}$, the roughness of the kernel is $R(K) := \int K(\mathbf{u})^2 d\mathbf{u}$. For simplicity, we assumed a Gaussian kernel and used the property $\int \mathbf{u}\mathbf{u}^T K(\mathbf{u}) d\mathbf{u} = I$. The kernel used for the derivation satisfies $\int K(\mathbf{u}) d\mathbf{u} = 1$ and $\int \mathbf{u} K(\mathbf{u}) d\mathbf{u} = 0$. With the derived LOMSE in Eq. (3), Kallus and Zhou [2] computed the bandwidth that minimizes the LOMSE as in Eq. (4). In the process, they assumed that $h \to 0$ and $\frac{1}{Nh^D} \to 0$ as $N \to \infty$.

$$h^* = \arg\min_h \text{LOMSE}(h, N, D) = \left(\frac{DC_v}{N4C_b}\right)^{\frac{1}{D+4}}, \tag{4}$$

2.3 Mahalanobis Distance Metric

Although relaxing the Dirac-delta target policy π to a kernel K makes importance sampling possible in our problem setting, the kernel is isotropic, and similarity of target and behavior actions in all directions are treated equally in weighing the reward in Eq. (2). But the similarities between two actions in some directions can be ignored relative to the other directions. Figure 1 illustrates the case where a Mahalanobis metric is locally learned at the target action for bias reduction in making a kernel-relaxed IS estimate. The bias of the estimate is reduced by altering the shape of the kernel to produce a higher similarity measure on a behavior action that has a similar reward to the target action.

The Mahalanobis distance metric can shorten or lengthen distances in certain directions. We define the Mahalanobis distance between two vectors $a_i \in \mathbb{R}^D$ and $a_j \in \mathbb{R}^D$ with the metric A as Eq. (5). Applying the Mahalanobis distance metric A $(= LL^T)$ to a kernel function can be seen as linearly transforming the kernel inputs with the transformation matrix L $(L^T a = z)$.

$$\|\mathbf{a}_i - \mathbf{a}_j\|_A = \sqrt{(\mathbf{a}_i - \mathbf{a}_j)^T A (\mathbf{a}_i - \mathbf{a}_j)} = \|\mathbf{z}_i - \mathbf{z}_j\|, \ (A \succ 0, \ A^T = A, \ |A| = 1) \tag{5}$$

(a) $K(\boldsymbol{a}, \boldsymbol{a_t}) = \frac{1}{2\pi} \exp\left(-\frac{(\boldsymbol{a}-\boldsymbol{a_t})^2}{2}\right)$ (b) $K(\boldsymbol{a}, \boldsymbol{a_t}; \boldsymbol{A}) = \frac{1}{2\pi} \exp\left(-\frac{(\boldsymbol{a}-\boldsymbol{a_t})^T A(\boldsymbol{a}-\boldsymbol{a_t})}{2}\right)$

Figure 1: Illustration of bias reduction in a IS estimate by local metric learning. Although behavior actions a_1 and a_2 are away from target action a_t by an equal euclidean distance, their corresponding rewards are different. When an (a) isotropic Gaussian kernel is used, bias can come from a_1 since the similarity measures of a_1, and a_2 from a_t are the same even though their rewards are different. However, when a (b) Mahalanobis distance metric is applied, the kernel shape can be altered to have a high similarity on a_2 that has a similar reward to a_t and have a low similarity measure on a_1 to avoid bias in kernel-relaxed IS.

3 Local Metric Learning for Kernel-Relaxed IS

To minimize the LOMSE of the kernel-relaxed IS estimate by learning a metric, we first analyze the LOMSE when the optimal bandwidth derived by Kallus and Zhou [2] is applied to the kernel. The optimal bandwidth presented in Eq. (4) balances between the leading bias and the leading variance in first and the second term of Eq. (3), respectively. However the balancing is conducted with an isotropic kernel and estimation bias that could have been avoided with proper use of kernel metric could be added in exchange of reduction in estimation variance as we see in Figure 1 which can worsen in a high dimensional space. To analytically show this, we adopt Proposition 4 in the work of Noh et al. [16] and show that the C_b contained in the leading order bias becomes a dominant term in the LOMSE of kernel-relaxed IS OPE given an optimal bandwidth in Eq. (4) and as the action dimension increases.

Proposition 1. *For action dimension $D \gg 4$ and given h^* (Eq. (4)), C_b contained in the leading bias in Eq. (3) is the dominant term of* $\mathrm{LOMSE}(h^*, N, D)$ *(Eq. (3)). By plugging in Eq. (4) to Eq. (3), we get the following relation for an environment with high action dimension:*

$$\mathrm{LOMSE}(h^*, N, D) = N^{-\frac{4}{D+4}} \left(\left(\frac{D}{4}\right)^{\frac{4}{D+4}} + \left(\frac{4}{D}\right)^{\frac{D}{D+4}} \right) C_b^{\frac{D}{D+4}} C_v^{\frac{4}{D+4}} \approx C_b. \tag{6}$$

(The proof is provided in Appendix A.3.)

The proposition 1 implies that in high action dimensional environment, MSE can be significantly reduced by the bias reduction. Therefore, we aim to reduce the leading estimation bias in Eq. (3) regardless of the bandwidth used in kernel-relaxed IS by using the Mahalanobis distance metric that shortens the distance between target and behavior actions in the direction where their corresponding rewards are similar.

As mentioned in Section 2.3, applying a metric ($A = LL^T$) to a kernel is equivalent to linearly transforming input vectors of the kernel. Therefore, the leading order bias with a metric should be analyzed with linearly transformed actions ($\mathbf{z} = L^T \mathbf{a}$). By applying Taylor expansion on $p(r|s, a)$ and using change of variable, we get: (derivation in A.5.).

$$\begin{aligned}
\mathrm{Bias}\left[\hat{\rho}^K\right] =& \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}), \mathbf{a}\sim\pi_b(\mathbf{a}\,|\,\mathbf{s}), r\sim p(r|\,\mathbf{s},\mathbf{a})}\left[\hat{\rho}^K - \rho^\pi\right] && (7)\\
\approx& \frac{h^2}{2} \iint r p(\mathbf{s})\, \mathrm{tr}\left(\mathbf{H_z}\, p(r \mid \mathbf{s}, \mathbf{z})|_{z=L(s)^T\pi(\mathbf{s})}\right) dr d\mathbf{s} && (8)\\
=& \frac{h^2}{2} \mathbb{E}_{s\sim p(s)}\left[\mathrm{tr}\left(A(\mathbf{s})^{-1}\, \mathbf{H_a}\, r(\mathbf{s}, \mathbf{a})\big|_{a=\pi(\mathbf{s})}\right)\right] && (9)
\end{aligned}$$

where, $\mathbf{H}$ is the Hessian operator and the expected reward is $r(s, a) := \mathbb{E}[r|s, a]$. In the derivation, we assumed using a Gaussian kernel and used the property $\int \mathbf{u}\mathbf{u}^T K(\mathbf{u}) d\mathbf{u} = I$. From Eq. (9), the leading order bias in Eq. (9) which compose LOMSE can be minimized with $A(\mathbf{s})$. In the following

Theorem 1, we introduce the optimal metric matrix $A_{opt}(\mathbf{s})$ that minimizes the squared leading order bias by using the semi-definite programming solution presented in the work of Noh et al. [17] on the leading order bias where a kernel metric is applied in Eq. (9)

Theorem 1. *Under the assumption that the $p(r|s,a)$ is twice differentiable w.r.t. an action $\mathbf{a}$, mahalanobis metric can be applied to reduce the absolute value of the leading order bias in Eq.* (3).

when $\Lambda_+(\mathbf{s})$ and $\Lambda_-(\mathbf{s})$ are diagonal matrices of positive and negative eigenvalues of the Hessian $\mathbf{H_a}\,\mathbb{E}[r|\,\mathbf{s},\mathbf{a}]|_{\mathbf{a}=\pi(\mathbf{s})}$, $U_+(\mathbf{s})$ and $U_-(\mathbf{s})$ are matrices of eigenvectors corresponding to $\Lambda_+(\mathbf{s})$ and $\Lambda_-(\mathbf{s})$, respectively. And $d_+(\mathbf{s})$ and $d_-(\mathbf{s})$ are the numbers of positive and negative eigenvalues of the Hessian, then the metric $A_{opt}(\mathbf{s})$ that minimizes the squared leading order bias is proportional to:

$$A_{opt}(\mathbf{s}) \propto [U_+(\mathbf{s})U_-(\mathbf{s})] \begin{pmatrix} d_+(\mathbf{s})\Lambda_+(\mathbf{s}) & 0 \\ 0 & -d_-(\mathbf{s})\Lambda_-(\mathbf{s}) \end{pmatrix} [U_+(\mathbf{s})U_-(\mathbf{s})]^T \tag{10}$$

The derivation of the metric involves solving a Lagrangian equation to minimize the absolute value of the trace term in Eq. (9) *with constraints on $A_{opt}(s)$ (Detailed proof is provided in Appendix A.6.)*

The derived metric matrix proportional to $A_{opt}(\mathbf{s})$ can be divided by its determinant to be $A_{opt}(\mathbf{s})$. And in the special case where the Hessian matrix contains both positive and negative eigenvalues, the metric reduces the leading order bias in Eq. (3) to zero (see Appendix A.6). $A_{opt}(\mathbf{s})$ is locally computed at a state $\mathbf{s}$ and target action $\pi(\mathbf{s})$ as it is derived from the Hessian at that point $\mathbf{H_a}\,r(s,a)|_{a=\pi(s)}$. The eigenvalues and eigenvectors of the Hessian matrix represent the principal direction of the expected reward curvatures and principal curvatures in the corresponding directions. The principal directions of curvatures are the directions where the expected reward has the largest curvatures. When the metric learned from the Hessian is applied to the kernel, it can be seen as using Mahalanobis distance instead of Euclidean distance, as shown in Figure 1 and in Eq. (11):

$$\|\mathbf{a}-\pi(\mathbf{s})\|^2_{A_{opt}} = \sum_{i=1}^{d_+}\left\{\sqrt{d_+\lambda_{+,i}}\mathbf{u}^T_{+,i}(\mathbf{a}-\pi(\mathbf{s}))\right\}^2 + \sum_{i=1}^{d_-}\left\{\sqrt{-d_-\lambda_{-,i}}\mathbf{u}^T_{-,i}(\mathbf{a}-\pi(\mathbf{s}))\right\}^2 \tag{11}$$

where, $\mathbf{u}_{+,i}$ and $\mathbf{u}_{-,i}$ are positive and negative eigenvectors, respectively, and their corresponding eigenvalues are denoted as $\lambda_{+,i}$ and $\lambda_{-,i}$, respectively. The Mahalanobis distance is formed to increase the distance between the behavior action $\mathbf{a}$ and the target action $\pi(\mathbf{s})$ on the expected reward's principal directions of curvatures weighted by $\sqrt{d_+\lambda_{+,i}}$ or $\sqrt{-d_-\lambda_{-,i}}$. To sum up, the derived $A_{opt}(\mathbf{s})$ is locally learned at state $\mathbf{s}$ and corresponding target action $\pi(\mathbf{s})$ in a way that increases the Mahalanobis distance between a behavior action $\mathbf{a}$ and a target action $\pi(\mathbf{s})$ on the directions where the expected reward's rate of change is the steepest.

We showed that by applying our metric to a kernel-relaxed IS estimator, the bias of the IS estimation can be reduced, and therefore, the MSE is reduced. Next, we see how the KMIS metric affects the convergence rate of a kernel-relaxed IS estimator w.r.t. datasize N and the action dimension D similar to Theorem 3 in the work of Kallus and Zhou [1].

Theorem 2. *Kernel-relaxed IS estimator with the KMIS metric and the optimal bandwidth h^* in Eq.* (4) *is a consistent estimator in which convergence rate is equal or faster than the isotropic kernel-relaxed IS estimator with h^*. The KMIS applied estimator converges to the true policy value faster than the one without the metric by the rate of $O(D^{-\frac{1}{2}})$ as the action dimension increases.*

The proof is made by analyzing the complexity of the sample size N and the action dimension D of the LOMSE in Eq. (3) *with and without the proposed metric. (The proof is provided in Appendix A.7.)*

Since the derived optimal bandwidth $A_{opt}(\mathbf{s})$ in Theorem 1 assumes that $p(r|s,a)$ is twice differentiable, we add in a regularizers $\beta(\mathbf{s})$ and $\gamma(\mathbf{s})$ as in Eq. (12) so that the metric applied kernel can fall back to an isotropic kernel in the case where the assumption is not met in practice. The regularizers are also adjusted to satisfy $|A| = 1$ (See Appendix A.8 for more detail). For the simplicity of the algorithm, we did not discard the components along the eigenvectors with zero eigenvalues (U_0) but let the metric have a longer relative bandwidth on the directions compared to the others via regularizers.

$$A_{MIS}(\mathbf{s}) = \beta(\mathbf{s})\,[U_+U_-U_0] \begin{pmatrix} d_+\Lambda_+ & 0 & 0 \\ 0 & -d_-\Lambda_- & 0 \\ 0 & 0 & \mathbf{0} \end{pmatrix} [U_+U_-U_0]^T + \gamma(\mathbf{s})\mathbb{I} \tag{12}$$

For the implementation of the proposed local metric A_{MIS} for kernel-relaxed IS estimation, we fit a neural network based reward regressor with a Gaussian output layer. The regressor learns the mean and the variance of the reward distribution from data which can be used for estimating the Hessian required for our algorithm. The reward regressor may have some error regarding the estimation of the Hessian and could affect the estimation error of our algorithm. We analytically analyze how the error in the reward regressor (error in Hessian estimation) can affect the estimation error and the convergence speed of our algorithm in Appendix A.10). And we empirically test if our metric is still accurate in case where noise in the reward data increases and degrades the estimation performance of DM in Appendix A.4. The reward regressor also can be used to estimate the optimal bandwidth. After obtaining the Hessian from the regressor, we linearly transform the kernel inputs with the transformation matrix L from the proposed metric $A_{MIS} = LL^T$ as in Eq. (13). We provide pseudo code in Algorithm 1 for a better understanding of the readers.

$$L(\mathbf{s}) = [U_+ U_- U_0] \left[\begin{pmatrix} \beta(\mathbf{s}) d_+ \Lambda_+ & 0 & 0 \\ 0 & -\beta(\mathbf{s}) d_- \Lambda_- & 0 \\ 0 & 0 & \mathbf{0} \end{pmatrix} + \gamma(\mathbf{s}) \mathbb{I} \right]^{\frac{1}{2}} \tag{13}$$

Algorithm 1 KMIS for Kernel-Relaxed IS Estimation

Input: offline data $D = \{\mathbf{s}_i, \mathbf{a}_i, r_i\}_{i=1}^N$, behavior policy π_b, deterministic target policy π, reward function parameters ϕ, bandwidth h, kernel K.

Output: Estimate of the target policy value $\hat{\rho}^K$

1: Fit the neural network regressor $r_\phi(\mathbf{s}, \mathbf{a})$ with D
2: Compute $\mathbf{H_a}\, r_\phi(\mathbf{s}, \mathbf{a})|_{a=\pi(\mathbf{s})}$
3: Obtain eigenvectors and eigenvalues of $\mathbf{H_a}\, r_\phi(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$
4: Compute L(s) as in Eq. (13)
5: Transform the kernel inputs: $\mathbf{z} = L(\mathbf{s})^T (\pi(\mathbf{s}) - \mathbf{a})$
6: Compute the estimate: $\frac{1}{Nh^D} \sum_{i=1}^N K\left(\frac{\mathbf{z}}{h}\right) \frac{r_i}{\pi_b(\mathbf{a}_i|\mathbf{s}_i)}$

4 Experiments

The experiments show that KMIS can be used with the existing works on bandwidth learning, and their MSEs are reduced by our metric. Since our algorithm learns the metric for kernel-relaxed IS, we compare our algorithm with the other kernel based IS methods. The baselines include a discretized OPE estimator [2] that discretizes the action space by 10 for each action dimension (100 bins) for an IS estimation (for details, see Appendix A.9). The work of Kallus and Zhou [2] and SLOPE [15] are also compared, where both learn bandwidths for kernel-relaxed IS. For both baselines, we use the code provided by the authors. For Kallus and Zhou's estimator, we use a reward regressor for for bandwidth learning instead of making kernel density estimation (KDE) as in their work since using KDE to estimate the bandwidth is computationally expensive. KMIS uses both the regressor and the learned bandwidth by the baselines for OPE. Kallus and Zhou's estimator and our KMIS share the same neural network reward regressor composed of two hidden layers of 128 neurons for learning bandwidth and metric, respectively. For the activation function of the neural network, we used tanh. For more details and codes for our algorithm, look at Appendix A.9. For SLOPE, we select a bandwidth from geometrically spaced bandwidths $\left\{2^{-i} : i \in [1, 7], i \in \mathbb{Z}\right\}$ as in their work [15] for the synthetic domain. For the Warfarin data, we selected a bandwidth from $\left\{2^{-i} : i \in [-2, 7], i \in \mathbb{Z}\right\}$. We do not correct the boundary bias as in the works of Kallus and Zhou [2], Su et al. [15] since we regard the bias induced by the boundaries are negligible compared to the bias reduced by our locally learned metric. We test our algorithm and baselines on various synthetic domains and the Warfarin dosing dataset provided by International Warfarin Pharmacogenetics. [23].

4.1 Synthetic Data

In the experiment with the synthetic data, we show that our metric can be applied to offline data sampled from environments with various reward functions. Furthermore, we show that learning local metrics can reduce MSEs of existing kernel-relaxed IS estimators. To show this, three kinds of synthetic test domains are prepared.

We first test on the synthetic domain with a quadratic reward function w.r.t. actions. As our metric is computed from the Hessian $\mathbf{H_a}\, r(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$, we first test if the metric can be learned from the neural network reward regressor. If the metric is learned successfully, the metric would be formed in an ellipsoidal shape and would be the same for every state and target action as the Hessian of a quadratic function is a constant. In the experiment, although our estimator uses the Hessian information from the neural network reward regressor, our metric learning applied kernel-relaxed IS estimator outperforms the reward regressor, and other baselines.

After checking if the metric can be properly learned from the reward regressor, we see if the metric can still be learned in an environment where negative absolute error is given as a reward. Since the Hessian of the reward function is zero at the reward, it is reasonable to think that the metric learning would fail in such a case. However, since we are dealing with finite samples and because we are using a neural network reward regressor, we conjecture that some form of convex shaped reward estimate is learned by the regressor and makes metric learning possible. We also let one of the action dimensions be unrelated to the reward. If metric learning is successful, the metric will be learned to have a relatively larger bandwidth to the direction of the action dimension.

Lastly, we test KMIS on an exponential reward whose shape is similar to the rewards used in multi-goal environments, which was introduced in the works of Haarnoja et al. [24]. Since the two synthetic test domains previously introduced have true Hessian $\mathbf{H_a}\, r(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$ of constant and zero, respectively, we test our algorithm on an exponential reward where it has Talyor expansion terms larger than the second order.

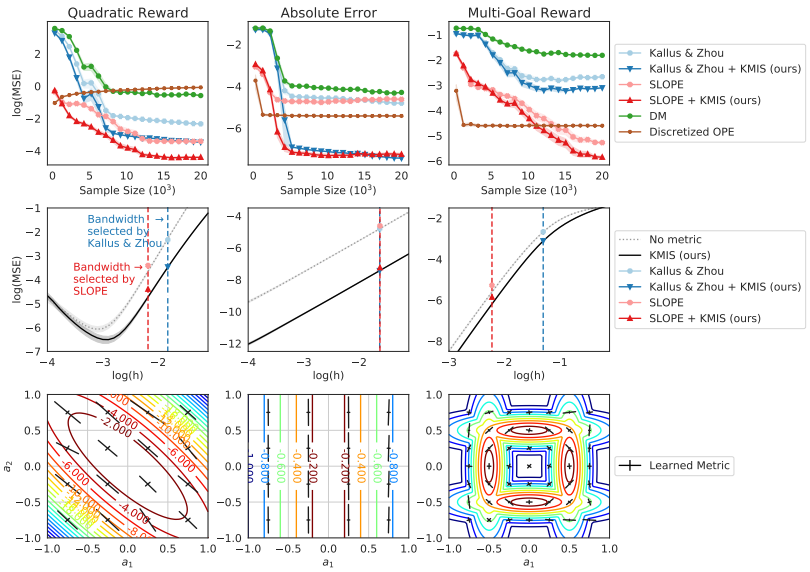


Figure 2: Experimental results on the synthetic data. Columns starting from the left are experimental results from the test domain of quadratic reward, negative absolute error with dummy action dimension, Multi-Goal reward made with the exponential functions. The first row shows how estimations are made as more data is acquired. Means and standard errors of squared errors obtained from 100 trials are drawn in the figures. The second row shows how the metrics reduce the MSEs at given bandwidths. Means and standard errors of squared errors obtained from 80 trials are drawn in the figures. The second row also marks the results of the first row when the datasize used in the estimation is 20k. The last row shows the learned metric at each test domain.

All synthetic test domains use actions and states that are $\mathbf{a} \in \mathbb{R}^2$, and $\mathbf{s} \in \mathbb{R}^2$. For the test domain with the quadratic reward, the states are uniformly sampled in the range of $[-1, 1]$, behavior policy is the Gaussian $\pi_b(a|s) = N(\mathbf{s} + \left[\begin{smallmatrix}0.2\\0.2\end{smallmatrix}\right], 0.5^2\mathbb{I})$, target policy is $\pi(\mathbf{s}) = s$, and the reward is a quadratic function w.r.t. actions. For the details, look at Appendix A.9. The test environment with the negative absolute error is prepared by having both states and actions uniformly sampled in the range of $[-1, 1]$, the target policy is designed to be $\pi(\mathbf{s}) = 0.5\mathbf{s}$, and the reward is $r(\mathbf{s}, \mathbf{a}) = -|0.5s_1 - a_1|$, where s_1 and a_1 are the first dimensions of the state and action vectors. The test domain with the exponential

reward is made with states and actions uniformly sampled from the range of $[-1, 1]$, and target policy $\pi(\mathbf{s}) = \mathbf{s} + \left[\begin{smallmatrix}0.5\\0\end{smallmatrix}\right]$. For the details of the exponential reward, look at Appendix A.9.

The experimental results in Figure 2 show that metric learning reduces the MSEs in the kernel relaxed IS estimates and also outperforms the reward regressor, which is used for computing the Hessian required for learning the metric. The results in the first row of Figure 2 show that the MSE of SLOPE and Kallus et al.'s estimator are reduced by our metric. To see whether the learning metrics always reduce MSE in the kernel-relaxed IS estimator, we test KMIS on given bandwidths. Seeing the results in the second row of Figure 2, we can see that most of the time, the MSE is reduced with metric learning. For the discretized OPE estimator, it performs worst in the quadratic reward domain. But for the other domains, it performs better than the other baselines. The absolute error domain and the multi-goal domain have uniform action distributions and no noise in the rewards. In this condition, because we apply self-normalization to all estimators, the discretized OPE would take an average of the behavior rewards which corresponding behavior actions are in the same discretized bins with the target actions which can be accurate in this condition. In the second row of Figure 2, the dotted vertical lines present the average of the bandwidths learned by Kallus et al. (blue) or SLOPE (red), and the markers show the average of the learned bandwidths and the MSE by each estimator when 20k data points are used to obtain the OPE result. Since the learned bandwidth can vary for each run, the markers may not exactly lie on the black lines denoting the fixed bandwidth's results. To see how metric is learned and also to see how it helps reduce the bias of the kernel-relaxed IS estimate, we show the reward landscape at $\mathbf{s} = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ and the learned metric in black crosses in the last row. By looking at the metric learned for the quadratic reward, we see that our metric is properly learned with the Hessian $\mathbf{H_a}\, r(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$ provided by the reward regressor as the learned metrics are similar for most actions. For the negative absolute error test domain with the dummy action dimension, we see that the metric is learned to ignore the dummy action dimension a_2 by having longer relative bandwidth to the direction of a_2 while having shorter relative bandwidth in the direction of a_1 which is used for computing the reward. This result empirically verifies our conjecture made earlier, which suggests that our metric learning algorithm can be generally applied to the dataset where the expected reward or $p(r|s, a)$ is not twice differentiable. The metric learned on the multi-goal reward made with exponents verifies that our metric indeed learns the local metric for each action according to the reward landscape.

In Proposition 1, we discussed that the leading order bias becomes a dominant term in the LOMSE of isotropic kernel smoothed IS estimator. To validate the Proposition, we show the empirical bias-variance decomposition of MSEs ($\mathrm{MSE}[\widehat{\rho}^K] = \mathrm{Bias}[\widehat{\rho}^K]^2 + Var[\widehat{\rho}^K]$) of the Kallus&Zhou and Kallus&Zhou+KMIS estimates in Figure .5. Results show that empirical estimation bias increases as the action dimension increases and both algorithms suffer from high bias. But our method reduces bias with the learned metric and shows best performance. While Kallus&Zhou+KMIS has reduced empirical bias compared to Kallus&Zhou, its variance is increased more than that of Kallus&Zhou. Although the leading order variance is unaffected by the metric, the ignored components of the estimation variance could be affected by the metric and resulted in increase in the variance.

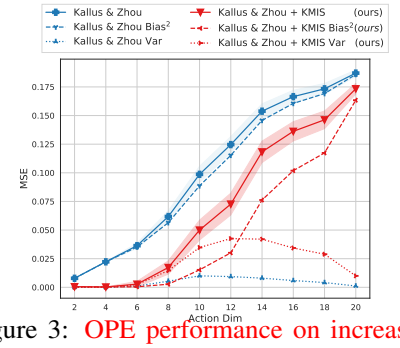
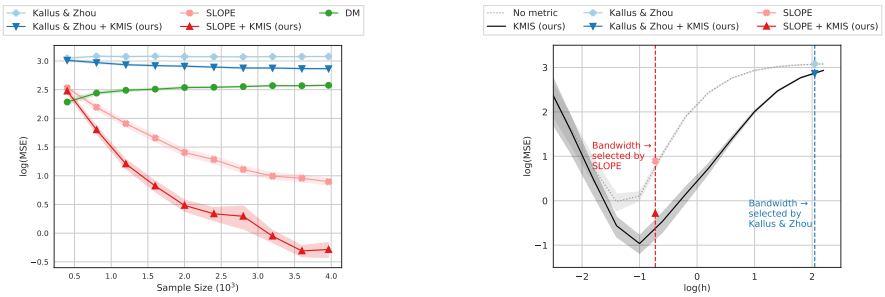


Figure 3: OPE performance on increasing number of action dimensions in the absolute error domain. Means and standard errors of squared errors obtained from 100 trials with 20k samples are drawn in the figures. Empirical bias and variance of the estimates are also drawn with the same estimates.

4.2 Warfarin Dosing Data

To test our algorithm on a more realistic dataset, we test our algorithm on the Warfarin dosing dataset [23]. Warfarin is a treatment that is commonly used for preventing blood clots. The dataset contains information on the patients who got treated, therapeutic doses, and the resulting outcomes. The outcomes of the treatments are reported in the international normalized ratio (INR), which measures how quickly blood clots. We use a similar experimental setting used by Kallus and Zhou [2]. As in their work, we use the 81 features of the patient information selected by Kallus and Zhou [2] as states $\mathbf{s}$. For the behavior action vectors, the first dimension is sampled from the normal distribution



(a) (b)

Figure 4: Experimental results on the Warfarin Data. (a) Performance of estimators on the Warfarin Data. The experiment is repeated in 300 trials, and MSE and standard error of the squared error are reported. (b) Reduction of MSE by metric at given bandwidths. The experiment is repeated in 100 trials. MSE and standard error of the squared error are reported. The dotted vertical lines present the average of the bandwidths learned by Kallus et al. (blue) or SLOPE (red), and the markers show the average of the learned bandwidths and the MSE by each estimator when all 3964 samples are used to obtain the OPE result. Since the learned bandwidth can vary for each run, the markers may not exactly lie on the black lines denoting the fixed bandwidth's results.

$N(\mu^* + \sigma^*\sqrt{0.5}z_{BMI}, (\sigma^*\sqrt{0.5})^2)$ truncated by the minimum and maximum therapeutic doses $a^*_{\min}$ and $a^*_{\max}$ in the data. z_{BMI} is the z score of patients' BMIs, μ^* and σ^* are the mean and the standard deviation of the therapeutic dosages, respectively. To test our algorithm we add a dummy action dimension sampled from a uniform distribution $a_2 \sim \text{unif}[a^*_{\min}, a^*_{\max}]$. For the reward, since Kallus and Zhou [2] reported that the INR in the dataset is inadequate for testing OPE algorithms, we use their cost function as a negative reward function $r = -\max(|a_1 - a^*| - 0.1a^*, 0)$. The reward function is only dependent on the first action dimension. The target policy is $\pi(s) = \left[\begin{smallmatrix} s_{BMI} \\ 0 \end{smallmatrix}\right]$.

Experimental results in Figure 4a show that our metric reduces the MSE of the kernel-relaxed IS estimate in Warfarin data that is more realistic than that of the synthetic domain. In the figure, we see that even though the MSE of the reward regressor is high, SLOPE + KMIS is able to learn the metric that reduces the MSE of SLOPE. Kallus and Zhou + KMIS also reduce the MSE of the estimator proposed by Kallus and Zhou. However, since the bandwidth learned by Kallus and Zhou's model is too big, as shown in Figure 4 there are limitations in the reduction of its MSE via metric.

5 Discussion and Conclusion

We present local metric learning for kernel-relaxed IS (KMIS). KMIS locally learns metrics at a given state, and the metric reduces the bias of the kernel-relaxed IS estimate. This is done by learning the metric to assign high similarity to the behavior action that has similar rewards to that of the target action for IS. Specifically, given a state, our proposed method learns the Mahalanobis distance metric that shortens the distance between a pair of behavior and target actions when their corresponding rewards are similar. Our experimental results suggest that applying KMIS to a kernel-relaxed IS estimation reduces MSE in most cases regardless of given bandwidths. Moreover, we also showed that KMIS could be used for offline data sampled from environments with various reward functions. And most importantly, KMIS can be used with other kernel-based IS estimators that learn bandwidths. The limitation of our work is that KMIS assumes that a optimal bandwidth is provided. Because of this, when suboptimal bandwidth is given, our algorithm may not fully function as intended. Another limitation is that our algorithm is dependent on the Hessian estimated by the reward regressor which can be erroneous. Although our work has some limitations, we have shown in the empirical studies that KMIS reduces MSEs of kernel-relaxed IS estimates for almost every bandwidths. Our work also opens a new research direction for the kernel-based methods. Our experimental results in Figure 2, especially the results from quadratic reward test domain, suggests that optimal bandwidths are different for a kernel with and without a metric. This result opens an interesting future study direction of jointly optimizing the bandwidth and the metric.

References

[1] Sheng Qiang and Mohsen Bayati. Dynamic pricing with demand covariates. *Available at SSRN 2765257*, 2016.

[2] Nathan Kallus and Angela Zhou. Policy evaluation and optimization with continuous treatments. In *International conference on artificial intelligence and statistics*, pages 1243–1251. PMLR, 2018.

[3] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM conference on recommender systems*, pages 31–39, 2018.

[4] Yuta Saito, Shunsuke Aihara, Megumi Matsutani, and Yusuke Narita. Open bandit dataset and pipeline: Towards realistic and reproducible off-policy evaluation. *arXiv preprint arXiv:2008.07146*, 2020.

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

[6] Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 1097–1104, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

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

[8] Lu Wang, Andrea Rotnitzky, Xihong Lin, Randall E Millikan, and Peter F Thall. Evaluation of viable dynamic treatment regimes in a sequentially randomized trial of advanced prostate cancer. *Journal of the American Statistical Association*, 107(498):493–508, 2012.

[9] Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, pages 1447–1456. PMLR, 2018.

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

[11] Cameron Voloshin, Hoang M Le, Nan Jiang, and Yisong Yue. Empirical study of off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1911.06854*, 2019.

[12] Doina Precup. *Temporal abstraction in reinforcement learning*. University of Massachusetts Amherst, 2000.

[13] Kyle Colangelo and Ying-Ying Lee. Double debiased machine learning nonparametric inference with continuous treatments. *arXiv preprint arXiv:2004.03036*, 2020.

[14] Hengrui Cai, Chengchun Shi, Rui Song, and Wenbin Lu. Deep jump learning for off-policy evaluation in continuous treatment settings. *Advances in Neural Information Processing Systems*, 34, 2021.

[15] Yi Su, Pavithra Srinath, and Akshay Krishnamurthy. Adaptive estimator selection for off-policy evaluation. In *International Conference on Machine Learning*, pages 9196–9205. PMLR, 2020.

[16] Yung-Kyun Noh, Masashi Sugiyama, Kee-Eung Kim, Frank Park, and Daniel D Lee. Generative local metric learning for kernel regression. *Advances in Neural Information Processing Systems*, 30, 2017.

[17] Yung-Kyun Noh, Byoung-Tak Zhang, and Daniel D Lee. Generative local metric learning for nearest neighbor classification. In *Advances in neural information processing systems*, pages 1822–1830, 2010.

[18] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936.

[19] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

[20] OV Lepskii. Asymptotically minimax adaptive estimation. i: Upper bounds. optimally adaptive estimates. *Theory of Probability & Its Applications*, 36(4):682–697, 1992.

[21] OV Lepskii. On a problem of adaptive estimation in gaussian white noise. *Theory of Probability & Its Applications*, 35(3):454–466, 1991.

[22] Oleg V Lepski and Vladimir G Spokoiny. Optimal pointwise adaptive methods in nonparametric estimation. *The Annals of Statistics*, 25(6):2512–2546, 1997.

[23] International Warfarin Pharmacogenetics Consortium. Estimation of the warfarin dose with clinical and pharmacogenetic data. *New England Journal of Medicine*, 360(8):753–764, 2009.

[24] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pages 1352–1361. PMLR, 2017.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

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]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (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]
   (b) Did you include complete proofs of all theoretical results? [Yes]
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]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (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]

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]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
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 Appendix

A.1 Bias of the Kernel-Relaxed IS Estimate w.r.t Bandwidths

Bias of the kernel-relaxed IS estimate w.r.t bandwidths as derived in the work of Kallus and Zhou [1]. Define $\mathbb{E}_{\pi_b}[\cdot] := \mathbb{E}_{\mathbf{s}\sim \mathrm{p}(\mathbf{s}),\mathbf{a}\sim\pi_\mathrm{b}(\mathbf{a}|\mathbf{s}),\mathrm{r}\sim\mathrm{p}(\mathrm{r}|\mathbf{s},\mathbf{a})}[\cdot]$, then the bias of a kernel-relaxed IS estimate is:

$$\begin{aligned}\mathrm{Bias}[\hat{\rho}^K] &= \mathbb{E}_{\pi_b}[\hat{\rho}^K] - \rho^{\pi_e} \\ &= \mathbb{E}_{\pi_b}\left[\frac{1}{Nh^D}\sum_{i=1}^{N} K\left(\frac{\mathbf{a}_i - \pi(\mathbf{s}_i)}{h}\right)\frac{r_i}{\pi_b(\mathbf{a}_i \,|\, \mathbf{s}_i)}\right] - \rho^{\pi}\end{aligned} \tag{14}$$

For the first term of Eq. (14),

$$\begin{aligned}&\mathbb{E}_{\pi_b}\left[\frac{1}{Nh^D}\sum_{i=1}^{N} K\left(\frac{\mathbf{a}_i - \pi(\mathbf{s}_i)}{h}\right)\frac{r_i}{\pi_b(\mathbf{a}_i \,|\, \mathbf{s}_i)}\right] \\ &= \frac{1}{h^D}\iiint p(\mathbf{s})p(r|\,\mathbf{s},\mathbf{a})K\left(\frac{\mathbf{a}-\pi(\mathbf{s})}{h}\right) r dr d\mathbf{a} d\mathbf{s} \\ &= \iiint rp(\mathbf{s})\left(p(r|\mathbf{s},\pi_e(\mathbf{s})) + \frac{h^2}{2}\mathbf{u}^T\,\mathbf{H}_\mathbf{a}\, p(r|\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi_e(\mathbf{s})}\,\mathbf{u} + O\left(h^4\right)\right) K(\mathbf{u}) d\mathbf{u} dr d\mathbf{s} \\ &= \rho^\pi + \frac{h^2}{2}\iint rp(s)\,\mathrm{tr}\left(\left[\int \mathbf{u}\,\mathbf{u}^T K(\mathbf{u}) d\,\mathbf{u}\right]\mathbf{H}_\mathbf{a}\, p(r|\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}\right) dr d\,\mathbf{s} + O\left(h^4\right) \\ &= \rho^\pi + \frac{h^2}{2}\iint rp(\mathbf{s})\,\nabla^2_\mathbf{a} p(r|\mathbf{s},\mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\, dr d\,\mathbf{s} + O\left(h^4\right)\end{aligned}$$

$$\therefore \mathrm{Bias}[\hat{\rho}^K] = \frac{h^2}{2}\iint rp(\mathbf{s})\,\nabla^2_\mathbf{a} p(r|\mathbf{s},\mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\, dr d\,\mathbf{s} + O\left(h^4\right) \tag{15}$$

where the following relations are used in the derivation,

$$\begin{aligned}&\int K(\mathbf{u}) d\mathbf{u} = 1, \\ &\int \mathbf{u} K(\mathbf{u}) d\mathbf{u} = 0, \\ &\kappa_2(K) := \int \mathbf{u}\mathbf{u}^T K(\mathbf{u}) d\mathbf{u} = \mathbb{I} \text{ (By design on } K\text{)}, \\ &\mathbf{u} := \frac{\mathbf{a}-\pi(\mathbf{s})}{h}, \\ &\mathrm{h}^D d\mathbf{u} = d\mathbf{a},\end{aligned}$$

Taylor expansion of $p(r|\mathbf{s},\mathbf{a})$ at $\mathbf{a} = \pi(\mathbf{s})$,

$$\begin{aligned}p(r|\,\mathbf{s},\mathbf{a}) = p(r|\,\mathbf{s},\pi(\mathbf{s})) &+ (\mathbf{a}-\pi(\mathbf{s}))^T \nabla_\mathbf{a} p(r|\,\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})} \\ &+ \frac{1}{2}(\mathbf{a}-\pi(\mathbf{s}))^T\,\mathbf{H}_\mathbf{a}\, p(r|\,\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}(\mathbf{a}-\pi(\mathbf{s})) + \dots\end{aligned}$$

using the relation $h\mathbf{u} = \mathbf{a} - \pi(\mathbf{s})$,

$$\begin{aligned}p(r|\,\mathbf{s},\mathbf{a}) = p(r|\,\mathbf{s},\pi(\mathbf{s})) &+ h\mathbf{u}^T \nabla_\mathbf{a} p(r|\,\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})} \\ &+ \frac{h^2}{2}\mathbf{u}^T\,\mathbf{H}_\mathbf{a}\, p(r|\,\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}\mathbf{u} + \dots\end{aligned}$$

A.2 Variance of the Kernel-Relaxed IS Estimate w.r.t Bandwidths

Variance of the kernel-relaxed IS estimate w.r.t bandwidths as derived in the work of Kallus and Zhou [1].

Define $\mathbb{E}_{\pi_b}[\cdot] := \mathbb{E}_{\mathbf{s}\sim \mathrm{p}(\mathbf{s}),\mathbf{a}\sim\pi_\mathrm{b}(\mathbf{a}|\mathbf{s}),\mathrm{r}\sim\mathrm{p}(\mathrm{r}|\mathbf{s},\mathbf{a})}[\cdot]$, then the variance of a kernel-relaxed IS estimate is:

$$\begin{aligned}
\mathrm{Var}[\hat{\rho}^K] &= \mathrm{Var}\left[\frac{1}{Nh^D}\sum_{i=1}^{N} K\left(\frac{\mathbf{a}_i - \pi(\mathbf{s}_i)}{h}\right)\frac{r_i}{\pi_b(\mathbf{a}_i \,|\, \mathbf{s}_i)}\right] \\
&= \frac{1}{N^2} \times N \times \mathrm{Var}\left[\frac{1}{h^D} K\left(\frac{\mathbf{a} - \pi(\mathbf{s})}{h}\right)\frac{r}{\pi_b(\mathbf{a} \,|\, \mathbf{s})}\right] \\
&= \frac{1}{N}\left\{\mathbb{E}_{\pi_b}\left[\left(\frac{1}{h^D} K\left(\frac{\mathbf{a} - \pi(\mathbf{s})}{h}\right)\frac{r}{\pi_b(\mathbf{a} \,|\, \mathbf{s})}\right)^2\right]\right. \\
&\qquad\qquad \left. - \left(\mathbb{E}_{\pi_b}\left[\frac{1}{h^D} K\left(\frac{\mathbf{a} - \pi(\mathbf{s})}{h}\right)\frac{r}{\pi_b(\mathbf{a} \,|\, \mathbf{s})}\right]\right)^2\right\}
\end{aligned} \tag{16}$$

The second term of the Eq. (16) is,

$$\frac{1}{N}\left(\mathbb{E}_{\pi_b}\left[\frac{1}{h^D} K\left(\frac{\mathbf{a} - \pi(\mathbf{s})}{h}\right)\frac{r}{\pi_b(\mathbf{a} \,|\, \mathbf{s})}\right]\right)^2 \tag{17}$$

$$= \frac{1}{N}\left(\mathrm{Bias}(\hat{\rho}^K) + \rho^\pi\right)^2 \tag{18}$$

$$= \frac{1}{N}\left[\rho^\pi + \frac{h^2}{2}\iint r p(\mathbf{s})\, \nabla^2_{\mathbf{a}} p(r|\mathbf{s},\mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\, dr d\,\mathbf{s} + O\left(h^4\right)\right]^2 \tag{19}$$

$$\therefore \frac{1}{N}\left(\mathbb{E}_{\pi_b}\left[\frac{1}{h^D} K\left(\frac{\mathbf{a} - \pi(\mathbf{s})}{h}\right)\frac{r}{\pi_b(\mathbf{a} \,|\, \mathbf{s})}\right]\right)^2 = O\left(\frac{1}{N}\right) \tag{20}$$

The first term of the Eq. (16) is,

$$\frac{1}{N}\mathbb{E}_{\pi_b}\left[\left(\frac{1}{h^D} K\left(\frac{\mathbf{a} - \pi(\mathbf{s})}{h}\right)\frac{r}{\pi_b(\mathbf{a} \,|\, \mathbf{s})}\right)^2\right] \tag{21}$$

$$= \frac{1}{N}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\iint r^2 \frac{p(r|\,\mathbf{s},\mathbf{a})}{h^{2D}\pi_b(\mathbf{a} \,|\, \mathbf{s})} K\left(\frac{\mathbf{a} - \pi(\mathbf{s})}{h}\right)^2 d\,\mathbf{a}\, dr\right] \tag{22}$$

By substituting $\mathbf{a} = h\,\mathbf{u} + \pi(\mathbf{s})$ $\Big($ which comes from $\mathbf{u} = \frac{\mathbf{a} - \pi(\mathbf{s})}{h}\Big)$,

$$= \frac{1}{N}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\iint r^2 \frac{K(\mathbf{u})^2}{h^D}\frac{p(r|\,\mathbf{s}, h\,\mathbf{u} + \pi(\mathbf{s}))}{\pi_b(h\,\mathbf{u} + \pi(\mathbf{s})|\,\mathbf{s})} d\,\mathbf{u}\, dr\right] \tag{23}$$

Let $g_r(\mathbf{s},\mathbf{a}) := \frac{p(r|\,\mathbf{s},\mathbf{a})}{\pi_b(\mathbf{a} \,|\, \mathbf{s})}$, and apply Taylor expansion at $\mathbf{a} = \pi(\mathbf{s})$,

$$\begin{aligned}
g_r(\mathbf{s},\mathbf{a}) = g_r(\mathbf{s},\pi(\mathbf{s})) &+ (\mathbf{a} - \pi(\mathbf{s}))^T \nabla_{\mathbf{a}} g_r(\mathbf{s},\mathbf{a})\,|_{a=\pi(\mathbf{s})} \\
&+ \frac{1}{2}(\mathbf{a} - \pi(\mathbf{s}))^T\, \mathbf{H}_{\mathbf{a}}\, g_r(\mathbf{s},\mathbf{a})\,|_{a=\pi(\mathbf{s})}\,(\mathbf{a} - \pi(\mathbf{s})) + O(h^3)
\end{aligned}$$

substitute $\mathbf{a} = h\,\mathbf{u} + \pi(\mathbf{s})$,

$$g_r(\mathbf{s}, h\,\mathbf{u} + \pi(\mathbf{s})) = g_r(\mathbf{s}, \pi(\mathbf{s})) + h\mathbf{u}^{\mathsf{T}} \nabla_a g_r(\mathbf{s}, \mathbf{a}) \mid_{a=\pi(\mathbf{s})} + \frac{h^2}{2} \mathbf{u}^T \mathbf{H}_{\mathbf{a}}\, g_r(\mathbf{s}, \mathbf{a}) \mid_{a=\pi(\mathbf{s})} \mathbf{u} + O(h^3) \tag{24}$$

Plugging in Eq. (24) to Eq. (23),

$$\begin{aligned}
&\frac{1}{N} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\iint r^2 \frac{K(\mathbf{u})^2}{h^D} \frac{p(r|\,\mathbf{s}, h\,\mathbf{u} + \pi_e(\mathbf{s}))}{\pi_b(h\,\mathbf{u} + \pi(\mathbf{s})|\,\mathbf{s})} d\,\mathbf{u}\, dr \right] \\
&= \frac{1}{Nh^D} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\iint r^2 K(\mathbf{u})^2 \left(g_r(\mathbf{s}, \pi(\mathbf{s})) + \frac{h^2}{2} \mathbf{u}^{\mathsf{T}} \mathbf{H}_{\mathbf{a}}\, g_r(\mathbf{s}, \mathbf{a}) \mid_{a=\pi(\mathbf{s})} \mathbf{u} + O(h^4) \right) d\,\mathbf{u}\, dr \right] \\
&= \frac{R(K)}{Nh^D} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\int_R \frac{r^2 p(r|\,\mathbf{s}, \mathbf{a} = \pi(\mathbf{s}))}{\pi_b(\mathbf{a} = \pi(\mathbf{s})|\,\mathbf{s})} dr \right] \\
&\qquad + \frac{1}{2Nh^{D-2}} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\int_R \int r^2 K(\mathbf{u})^2 \left(\mathbf{u}^T \mathbf{H}_{\mathbf{a}}\, g_r(\mathbf{s}, \mathbf{a}) \mid_{a=\pi(\mathbf{s})} \mathbf{u} + O(h^4) \right) d\,\mathbf{u}\, dr \right] \\
&\qquad \left(\text{where, } R(K) := \int K(\mathbf{u})^2 d\,\mathbf{u} \right) \\
&= \frac{R(K)}{Nh^D} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\int \frac{r^2 p(r|\,\mathbf{s}, \mathbf{a} = \pi(\mathbf{s}))}{\pi_b(\mathbf{a} = \pi(\mathbf{s})|\,\mathbf{s})} dr \right] + O\left(\frac{1}{Nh^{D-2}} \right)
\end{aligned} \tag{25}$$

From Eq. (20) and Eq. (25), the variance can be represented as,

$$\therefore \mathrm{Var}[\hat{\rho}^K] = \frac{R(K)}{Nh^D} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\frac{\mathbb{E}\left[r^2 |\,\mathbf{s}, \mathbf{a} = \pi(\mathbf{s}) \right]}{\pi_b(\mathbf{a} = \pi(\mathbf{s})|\,\mathbf{s})} \right] + O\left(\frac{1}{Nh^{D-2}} \right) \tag{26}$$

A.3 Proof of Proposition 1

We adopt Proposition 4 in the work of Noh et al. [2] which shows that the bias is a dominant term in the MSE of kernel regression to our problem setting in kernel-relaxed IS OPE. We show that the bias is also a dominant term in the MSE of kernel-relaxed IS OPE when the action dimension is much bigger than 4.

Proposition 1. *For action dimension $D \gg 4$ and given h^* (Eq. (4)), C_b contained in the leading bias in Eq. (3) is the dominant term of $\mathrm{LOMSE}(h^*, N, D)$ (Eq. (3)). By plugging in Eq. (4) to Eq. (3), we get the following relation for an environment with high action dimension:*

$$\mathrm{LOMSE}(h^*, N, D) = N^{-\frac{4}{D+4}} \left(\left(\frac{D}{4}\right)^{\frac{4}{D+4}} + \left(\frac{4}{D}\right)^{\frac{D}{D+4}} \right) C_b^{\frac{D}{D+4}} C_v^{\frac{4}{D+4}} \approx C_b. \tag{27}$$

Proof. The leading order MSE derived w.r.t. h by Kallus and Zhou [2] is,

$$\mathrm{LOMSE}(h, N, D) := h^4 C_b + \frac{C_v}{N h^D} \tag{28}$$

When using a kernel without a metric, the optimal bandwidth (h^*) that minimizes the leading order MSE is,

$$\frac{d}{dh}(\mathrm{LOMSE}(h, N, D)) = 4h^3 C_b - D h^{-D-1} N^{-1} C_v \tag{29}$$

$$4(h^*)^3 C_b - D(h^*)^{-D-1} N^{-1} C_v = 0 \tag{30}$$

$$h^* = \left(\frac{D C_v}{N 4 C_b}\right)^{\frac{1}{D+4}} \tag{31}$$

And the leading order MSE with the optimal kernel bandwidth is,

$$\mathrm{LOMSE}(h^*, N, D) = N^{-\frac{4}{D+4}} \left(\left(\frac{D}{4}\right)^{\frac{4}{D+4}} + \left(\frac{4}{D}\right)^{\frac{D}{D+4}} \right) C_b^{\frac{D}{D+4}} C_v^{\frac{4}{D+4}} \tag{32}$$

$$\approx C_b. \text{ (for } D \gg 4) \tag{33}$$

□

A.4 Experiment with Varying Noise Level in the Rewards

In Figure 5, we also observe increasing estimation bias when the noise in the rewards increases. DM suffers the most as the function estimation becomes difficult as the noise increases. However, our metric reduces leading order bias and shows best performance.

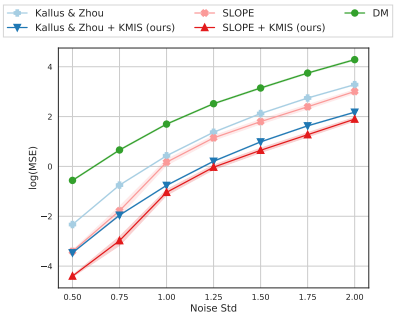


Figure 5: Performance of OPE algorithms with increasing standard deviation of a the Gaussian noise in the rewards in quadratic reward domain. Means and standard errors of squared errors obtained from 100 trials are drawn in the figures.

A.5 Derivation of Eq. (9)

For the derivation, we use the following relations:

$$
\begin{aligned}
p(r \mid \mathbf{s}, \mathbf{z}) &= \frac{p(r, \mathbf{s}, \mathbf{z})}{p(\mathbf{s}, \mathbf{z})} \\
&= \frac{p(r, \mathbf{s}, \mathbf{a}) \left|\frac{\partial \mathbf{a}}{\partial \mathbf{z}}\right|}{p(\mathbf{s}, \mathbf{a}) \left|\frac{\partial \mathbf{a}}{\partial \mathbf{z}}\right|} \\
&= p(r \mid \mathbf{s}, \mathbf{a})
\end{aligned} \tag{34}
$$

$$
\begin{aligned}
\mathbf{H}_{\mathbf{z}}\, p(r \mid \mathbf{s}, \mathbf{z})\big|_{\mathbf{z}=L(\mathbf{s})^T \pi(\mathbf{s})} &= \frac{\partial^2}{\partial \mathbf{z}\, \partial \mathbf{z}^T} p(r \mid \mathbf{s}, \mathbf{z}) \\
&= \frac{\partial}{\partial \mathbf{z}} \left(\frac{\partial}{\partial \mathbf{z}} p(r \mid \mathbf{s}, \mathbf{z}) \right)^T \\
&= \frac{\partial}{\partial \mathbf{z}} \left(L(\mathbf{s})^{-1} \frac{\partial}{\partial \mathbf{a}} p(r \mid \mathbf{s}, \mathbf{a}) \right)^T, \text{by Eq. (34)} \\
&= \frac{\partial \mathbf{a}}{\partial \mathbf{z}} \frac{\partial}{\partial \mathbf{a}} \left(\left(\frac{\partial}{\partial \mathbf{a}} p(r \mid \mathbf{s}, \mathbf{a}) \right)^T L(\mathbf{s})^{-T} \right) \\
&= L(\mathbf{s})^{-1} \frac{\partial^2}{\partial \mathbf{a}\, \partial \mathbf{a}^T} p(r \mid \mathbf{s}, \mathbf{a}) L(\mathbf{s})^{-T} \\
&= L(\mathbf{s})^{-1}\, \mathbf{H}_{\mathbf{a}}\, p(r \mid \mathbf{s}, \mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\, L(\mathbf{s})^{-T}
\end{aligned} \tag{35}
$$

Plugging in Eq. (35) to Eq. (8), and changing the ordering inside the trace operator,

$$
\begin{aligned}
\text{Bias}\left[\hat{\rho}^K\right]^2 &\approx \left(\frac{h^2}{2} \iint r p(\mathbf{s})\, \text{tr}\left(\mathbf{H}_{\mathbf{z}}\, p(r \mid \mathbf{s}, \mathbf{z})\big|_{\mathbf{z}=L(\mathbf{s})^T \pi(\mathbf{s})} \right) dr d\mathbf{s} \right)^2 \\
&= \left(\frac{h^2}{2} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\text{tr}\left(L(\mathbf{s})^{-1}\, \mathbf{H}_{\mathbf{a}}\, r(\mathbf{s}, \mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\, L(\mathbf{s})^{-T} \right) \right] \right)^2, \text{ by Eq. (35)} \\
&= \left(\frac{h^2}{2} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\text{tr}\left((L(\mathbf{s}) L(\mathbf{s})^T)^{-1}\, \mathbf{H}_{\mathbf{a}}\, r(\mathbf{s}, \mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})} \right) \right] \right)^2 \\
\therefore \text{Bias}\left[\hat{\rho}^K\right]^2 &\approx \left(\frac{h^2}{2} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\text{tr}\left(A(\mathbf{s})^{-1}\, \mathbf{H}_{\mathbf{a}}\, r(\mathbf{s}, \mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})} \right) \right] \right)^2
\end{aligned}
$$

A.6 Proof of Theorem 1

We use the semi-definite programming solution presented in the work of Noh et al. [3] for computing the metric matrix A_{opt} that minimizes the $\mathrm{tr}\left(A^{-1}(\mathbf{s})B(\mathbf{s})\right)^2$ for nearest neighbor classification. We use the solution to minimize the bias of kernel-relaxed IS OPE. In our case, we use $B(\mathbf{s}) := \mathbf{H}_{\mathbf{a}}\,\mathbb{E}[r|\,\mathbf{s},\mathbf{a}]|_{\mathbf{a}=\pi(\mathbf{s})}$ which comes from Eq. (8).

Theorem 1. *Under the assumption that the $p(r|\,\mathbf{s},\mathbf{a})$ is twice differentiable w.r.t. an action $\mathbf{a}$, when $\Lambda_+(\mathbf{s})$ and $\Lambda_-(\mathbf{s})$ are diagonal matrices of positive and negative eigenvalues of the Hessian $\mathbf{H}_{\mathbf{a}}\,\mathbb{E}[r|\,\mathbf{s},\mathbf{a}]|_{\mathbf{a}=\pi(\mathbf{s})}$, $U_+(\mathbf{s})$ and $U_-(\mathbf{s})$ are matrices of eigenvectors corresponding to $\Lambda_+(\mathbf{s})$ and $\Lambda_-(\mathbf{s})$, respectively. And $d_+(\mathbf{s})$ and $d_-(\mathbf{s})$ are the numbers of positive and negative eigenvalues of the Hessian, then the metric $A_{opt}(\mathbf{s})$ that minimizes the squared leading order bias is proportional to:*

$$A_{opt}(\mathbf{s}) \propto [U_+(\mathbf{s})U_-(\mathbf{s})] \begin{pmatrix} d_+(\mathbf{s})\Lambda_+(\mathbf{s}) & 0 \\ 0 & -d_-(\mathbf{s})\Lambda_-(\mathbf{s}) \end{pmatrix} [U_+(\mathbf{s})U_-(\mathbf{s})]^T \tag{36}$$

Proof. Define $B(\mathbf{s}) := \mathbf{H}_{\mathbf{a}}\, r(\mathbf{s},\mathbf{a})|_{\pi(\mathbf{s})}$, then we want $A(\mathbf{s})$ that,

$$\begin{aligned} &\min_{A(\mathbf{s})} \left(\mathrm{tr}\left(A^{-1}(\mathbf{s})B(\mathbf{s})\right)\right)^2 \\ &\text{s.t. } |A(\mathbf{s})| = 1 \\ &\quad A(\mathbf{s}) \succ 0 \\ &\quad A(\mathbf{s}) = A(\mathbf{s})^T \\ &\quad B(\mathbf{s}) = B(\mathbf{s})^T \end{aligned} \tag{37}$$

In the cases where some of the eigenvalues of $B(\mathbf{s})$ are 0, the components along those directions can be discarded.

When $B(\mathbf{s})$ has both positive and negative eigenvalues, $A_{opt}(\mathbf{s})$ makes the trace 0.

$$\begin{aligned} A_{opt}(\mathbf{s}) &= \beta(\mathbf{s})\,[U_+(\mathbf{s})U_-(\mathbf{s})] \begin{pmatrix} d_+(\mathbf{s})\Lambda_+(\mathbf{s}) & 0 \\ 0 & -d_-(\mathbf{s})\Lambda_-(\mathbf{s}) \end{pmatrix} [U_+(\mathbf{s})U_-(\mathbf{s})]^T \\ B(\mathbf{s}) &= [U_+(\mathbf{s})U_-(\mathbf{s})] \begin{pmatrix} \Lambda_+(\mathbf{s}) & 0 \\ 0 & \Lambda_-(\mathbf{s}) \end{pmatrix} [U_+(\mathbf{s})U_-(\mathbf{s})]^T \\ &\mathrm{tr}\left(A_{opt}^{-1}(\mathbf{s})B(\mathbf{s})\right) \\ &= \frac{1}{\beta(\mathbf{s})}\,\mathrm{tr}\left[[U_+(\mathbf{s})U_-(\mathbf{s})] \begin{pmatrix} \frac{\Lambda_+^{-1}(\mathbf{s})}{d_+(\mathbf{s})} & 0 \\ 0 & -\frac{\Lambda_-^{-1}(\mathbf{s})}{d_-(\mathbf{s})} \end{pmatrix} \begin{pmatrix} \Lambda_+(\mathbf{s}) & 0 \\ 0 & \Lambda_-(\mathbf{s}) \end{pmatrix} [U_+(\mathbf{s})U_-(\mathbf{s})]^T\right] \\ &= \frac{1}{\beta(\mathbf{s})}\left(\frac{d_+(\mathbf{s})}{d_+(\mathbf{s})} - \frac{d_-(\mathbf{s})}{d_-(\mathbf{s})}\right) \\ &= 0 \end{aligned}$$

where $\beta(\mathbf{s})$ is a scalar value dependent on a state $\mathbf{s}$ that scales the metric matrix to be $|A(\mathbf{s})| = 1$. When $B(\mathbf{s})$ only has negative or positive values, than we can solve the Lagrangian equation $F(\mathbf{s})$. In case where all eigenvalues of $B(\mathbf{s})$ are positive,

$$
\begin{aligned}
F(\mathbf{s}) &= \operatorname{tr}\left(A^{-1}(\mathbf{s})B(\mathbf{s})\right)^2 - c(\mathbf{s})(|A(\mathbf{s})| - 1) \\
\frac{\partial F(\mathbf{s})}{\partial c(\mathbf{s})} &= |A(\mathbf{s})| - 1 = 0 \\
\frac{\partial F(\mathbf{s})}{\partial A(\mathbf{s})} &= -2\operatorname{tr}\left(A^{-1}(\mathbf{s})B(\mathbf{s})\right) A^{-T}(\mathbf{s})B^T(\mathbf{s})A^{-T}(\mathbf{s}) - c(\mathbf{s})|A(\mathbf{s})|A^{-T}(\mathbf{s}) = 0 \\
c(\mathbf{s})\mathbb{I} &= -2\operatorname{tr}\left(A^{-1}(\mathbf{s})B(\mathbf{s})\right) A^{-1}(\mathbf{s})B(\mathbf{s}) \\
&\quad \left(\text{since } A(\mathbf{s}) = A^T(\mathbf{s}),\ B(\mathbf{s}) = B^T(\mathbf{s}),\ |A(\mathbf{s})| = 1\right)
\end{aligned} \tag{38}
$$

(i) Assume $c(\mathbf{s}) = 0$,

Then either $\operatorname{tr}\left(A^{-1}(\mathbf{s})B(\mathbf{s})\right) = 0$, or, $A^{-1}(\mathbf{s})B(\mathbf{s}) = \mathbf{0}$

We first check if $\operatorname{tr}\left(A^{-1}(\mathbf{s})B(\mathbf{s})\right) = 0$.

$$
\operatorname{tr}\left(A^{-1}(\mathbf{s})B(\mathbf{s})\right) = \operatorname{tr}\left(B^{\frac{1}{2}}(\mathbf{s})A^{-1}(\mathbf{s})B^{\frac{1}{2}}(\mathbf{s})\right) \tag{39}
$$

Since $B(\mathbf{s}) = U_+(\mathbf{s})\Lambda_+(\mathbf{s})U_+^T(\mathbf{s})$, $\quad B^{\frac{1}{2}}(\mathbf{s}) = B^{\frac{T}{2}}(\mathbf{s}) = U_+(\mathbf{s})\Lambda_+^{\frac{1}{2}}(\mathbf{s})U_+^T(\mathbf{s})$. And as $A(\mathbf{s}) \succ 0$, its inverse is also positive definite $A^{-1}(\mathbf{s}) \succ 0$. Using these relations, we can show that the term inside the trace in Eq. (39) is a positive definite matrix.

$$
\begin{aligned}
\mathbf{a}^T\left[B^{\frac{1}{2}}(\mathbf{s})A^{-1}(\mathbf{s})B^{\frac{1}{2}}(\mathbf{s})\right]\mathbf{a} &= \left(B^{\frac{T}{2}}(\mathbf{s})\mathbf{a}\right)^T A^{-1}(\mathbf{s})\left(B^{\frac{T}{2}}(\mathbf{s})\,\mathbf{a}\right) \\
&> 0, \quad \forall\, \mathbf{a} \in \mathbb{R}^D \backslash \{\mathbf{0}\}
\end{aligned}
$$

Because $B^{\frac{1}{2}}(\mathbf{s})A^{-1}(\mathbf{s})B^{\frac{1}{2}}(\mathbf{s}) \succ 0$, $\quad \operatorname{tr}\left(B^{\frac{1}{2}}(\mathbf{s})A^{-1}(\mathbf{s})B^{\frac{1}{2}}(\mathbf{s})\right) > 0$. Then, from Eq. (39),

$$
\begin{aligned}
\operatorname{tr}\left(A^{-1}(\mathbf{s})B(\mathbf{s})\right) &> 0 \\
A^{-1}(\mathbf{s})B(\mathbf{s}) &\neq \mathbf{0}, \quad \text{as it's trace is positive}
\end{aligned}
$$

Therefore, the assumption is wrong, and $c(\mathbf{s}) \neq 0$

(ii) Assume $c(\mathbf{s}) \neq 0$,

Since the Lagrangian multiplier $c(\mathbf{s})$ and $\operatorname{tr}\left(A^{-1}(\mathbf{s})B(\mathbf{s})\right)$ in Eq. (38) are scalar values, $A(\mathbf{s})$ needs to be a scalar multiple of $B(\mathbf{s})$ to match the scalar multiple of identity matrix in the LHS of Eq. (38). Therefore the $A_{opt}(\mathbf{s})$ is a scalar multiple of $B(\mathbf{s})$. Since $A(\mathbf{s})$ need to satisfy $|A(\mathbf{s})| = 1$, the optimal metric is,

$$
\therefore A_{opt}(\mathbf{s}) = \beta(\mathbf{s})U_+(\mathbf{s})\left[d_+(\mathbf{s})\Lambda_+(\mathbf{s})\right]U_+(\mathbf{s})^T \tag{40}
$$

where the $\beta(\mathbf{s})$ and $d_+(\mathbf{s})$ are scalars dependent on a state $\mathbf{s}$, and $\beta(\mathbf{s})$ scales the $A_{opt}(\mathbf{s})$ to be $|A_{opt}| = 1$.

Similarly, we can derive $A_{opt}(\mathbf{s})$ for the case where all eigenvalues of $B(\mathbf{s})$ are negative. □

A.7 Proof of Theorem 2

We compare the convergence speed of kernel-relaxed IS estimators with and without metrics learned by KMIS when both are provided with the optimal bandwidth by the work of Kallus and Zhou [1]. The convergence speed is analyzed similarly to Theorem 3 in the work of Kallus and Zhou [1].

Theorem 2. *Kernel-relaxed IS estimator with the KMIS metric and the optimal bandwidth h^* in Eq. (4) is a consistent estimator in which convergence rate is equal or faster than the isotropic kernel-relaxed IS estimator with h^*. The KMIS applied estimator converges to the true policy value faster than the one without the metric by the rate of $O(D^{-\frac{1}{2}})$ as the action dimension increases.*

Proof. The MSE of a kernel-relaxed IS estimation without a metric derived by Kallus and Zhou [1] in terms of action dimension D and data size of N is (from the derived bias and variance in Appendix A.1 and Appendix A.2),

$$\mathrm{MSE}(h, N, D) = \underbrace{h^4 C_b + O\left(h^6\right)}_{\mathrm{Bias}[\hat{\rho}^K]^2} + \underbrace{\frac{C_v}{N h^D} + O\left(\frac{1}{N h^{D-2}}\right)}_{\mathrm{Var}[\hat{\rho}^K]}, \tag{41}$$

where, the MSE g consists of squared bias and the variance of the estimate $\hat{\rho}^K$. Since the optimal bandwidth presented in Eq. (4) is $h^* = \mathcal{O}\left(\frac{D}{N}^{\frac{1}{D+4}}\right)$, the convergence of the MSE g without the metric is:

$$\begin{aligned}
\mathrm{MSE}(h^*, N, D) &= \mathcal{O}\left(\left(\frac{D}{N}\right)^{\frac{4}{D+4}}\right) + \mathcal{O}\left(N^{-1}\left(\frac{D}{N}\right)^{\frac{-D}{D+4}}\right) && (42)\\
&= \mathcal{O}\left(\left(\frac{D}{N}\right)^{\frac{4}{D+4}}\right) + \mathcal{O}\left(\left(\frac{D}{N}\right)^{\frac{4}{D+4}}\frac{1}{D}\right), \quad (D \geq 2) && (43)\\
&= \mathcal{O}\left(\left(\frac{D}{N}\right)^{\frac{4}{D+4}}\right) && (44)
\end{aligned}$$

Since the MSE of the kernel-relaxed IS estimator with h^* converges to 0 by $\mathcal{O}\left(\left(\frac{D}{N}\right)^{\frac{4}{D+4}}\right)$, The estimation approaches to the true policy value by the rate of $\mathcal{O}\left(\left(\frac{D}{N}\right)^{\frac{2}{D+4}}\right)$.

With our metric learning, C_b can be reduced. When the Hessian $\mathbf{H_a}\, r(s,a)|_{a=\pi(s)}$ contains both negative and positive eigenvalues, C_b is reduced to 0. And when the Hessian is a zero matrix, we cannot use metric and no bias can be reduced. Therefore, if we denote the C_b reduced by our metric as $C_{b,A}$,

$$C_b \geq C_{b,A} \geq 0 \tag{45}$$

In the best case, where $C_{b,A} = 0$, MSE with our metric and the optimal bandwidth $\mathrm{MSE}(h^*, A(\mathbf{s}), N, D)$ converges to zero by:

$$\begin{aligned}
\mathrm{MSE}(h^*, A(\mathbf{s}), N, D) &= (h^*)^4 \cancel{C_{b,A}} + O\left((h^*)^6\right) + \frac{C_v}{N (h^*)^D} + O\left(\frac{1}{N (h^*)^{D-2}}\right), && (46)\\
&= \mathcal{O}\left(\left(\frac{D}{N}\right)^{\frac{6}{D+4}}\right) + \mathcal{O}\left(\left(\frac{D}{N}\right)^{\frac{4}{D+4}}\frac{1}{D}\right) && (47)\\
&= \mathcal{O}\left(\left(\frac{D}{N}\right)^{\frac{4}{D+4}}\frac{1}{D}\right), (\text{ where, } N \gg D) && (48)
\end{aligned}$$

The MSE of our metric applied kernel-realxed IS estimation converges to 0 faster than the one without the metric by $\mathcal{O}(D^{-1})$ Therefore, our algorithm converges to the true policy value faster than the one without the metric by $\mathcal{O}(D^{-\frac{1}{2}})$.

$\square$

A.8 Regularizers for the KMIS Metric

To get $\beta(\mathbf{s})$ and $\gamma(\mathbf{s})$, we first derive the metric that is proportional to $A_{opt}(\mathbf{s})$ in Eq. (36). Then we add a small positive real coefficient $c(\mathbf{s})$ to it to avoid the metric matrix from having determinant of 0. The coefficient $c(\mathbf{s})$ is selected to be the maximum value among eigenvalues in the Hessian of reward multiplied by 0.01. We let the coefficient depend on the state since the scale of eigenvalues of a Hessian (of reward) matrix differs depending on experiment environments.

With the coefficient $c(\mathbf{s})$ added to the metric matrix, we compute the determinant of the resulting matrix $M(\mathbf{s})$. To scale the metric matrix to have determinant of 1, We scale the matrix by multiplying $\beta(s) = \frac{1}{|M|^{(-1/D)}}$ (where D is an action dimension). Then the $\gamma(s) = \beta(s) * c(s)$

A.9 Experiment Details

For all experiments, we used a neural network reward regressor with 2 fully-connected layers of 128 hidden units with tanh activations, and a Gaussian output layer. The neural network was trained with learning rate of 5e-4. The learning rate was chosen by conducting grid search over {1e-4, 5e-4, 1e-3}. The reward regressor was trained with early stopping rule where we stop training when the validation error does not decrease for 20 training epochs. And we used the trained weights that showed the lowest validation error during the training. For the train and validation split, 20% of the available offline data was used as the validation data, and the rest of the data was used for the training. For the estimation time, when we test our algorithm on the synthetic domain with absolute error and a dummy action dimension, our algorithm takes 40 seconds on average to make an OPE estimate with 20k offline data on the i7 CPU with 32GB RAM. The experimental results were made with 100 C2-standard-4 instances on the Google Cloud Platform where each instance has 4 virtual CPUs and 16GB of RAM.

Self-Normalization All estimators in the experiments are self-normalized as in the work of Kallus and Zhou [1]. Self-normalization is used because it reduces the estimation variance significantly with the addition of a small bias and yields a smaller MSE in result compared to the estimation without self-normalization [1, 4]. The self-normalized kernel-relaxed IS estimator without a metric is shown in Eq. (49).

$$\hat{\rho}^{K}_{\text{norm}} = \frac{\sum_{i=1}^{N} \frac{r_i}{\pi_b(\mathbf{a}_i \mid \mathbf{s}_i)} K\left(\frac{\mathbf{a}_i - \pi(\mathbf{s}_i)}{h}\right)}{\sum_{i=1}^{N} \frac{1}{\pi_b(\mathbf{a}_i \mid \mathbf{s}_i)} K\left(\frac{\mathbf{a}_i - \pi(\mathbf{s}_i)}{h}\right)} \tag{49}$$

No Boundary Bias Correction Action space can be bounded, and the kernels used in importance sampling can be extended past the bounds. As the actions of the offline data are only observed inside the bounds, the kernels extending outside the bounds will induce bias. Previous works [1, 5] made a correction for the induced bias by truncating the kernel by the action bounds of each dimension and normalizing the kernel. However, we do not correct the boundary bias for all estimators in this work since we regard the boundary bias as negligible compared to the bias reduced by our locally learned metric. Removing the boundary bias correction makes kernel-relaxed IS estimators simpler to use than those with corrections, as the implementation of boundary bias correction requires information on the action bounds, and the integration of the kernel within the action bounds.

Discretized OPE Estimator Discretized ope estimator discretizes the action space bounded by the minimum and maximum value of the behavior actions for each dimension with given number bins assigned for each dimension. In our experiment we discretized 10 for each dimension resulting in 100 discretized bins in 2D action space. The IS ratio of the discretized OPE estimator is composed of the indicator function in the numerator and the behavior policy probability density function integrated over the bin in the denominator.

A.9.1 Synthetic Domains Experiments

For the synthetic domain, we used dropout after each hidden layer with dropout rate of 0.5. The dropout rate was chosen by conducting grid search over {0.5, 0.8}. Behavior policy density values were clipped below 032 for the quadratic reward domain. The clipping threshold is chosen by conducting grid search over {3e-1, 1e-1, 1e-2}. For absolute error domain and multi-goal reward domain, as their behavior policy density is uniform density, we did not apply behavior density value clipping. The clipping threshold that showed stable convergence and low MSE was chosen. Clipping the behavior policy density value adds a small bias to the estimation, but reduces a significant variance, and therefore MSE is smaller compared to the estimations without the clipping [1].

Quadratic Reward Domain The conditional reward distribution given a state $\mathbf{s}$ and a action $\mathbf{a}$ is:

$$p(r|\,\mathbf{s}, \mathbf{a}) = N\left(r(\mathbf{s}, \mathbf{a}), 0.5^2\right), \tag{50}$$

where, $r(\mathbf{s}, \mathbf{a}) := \mathbb{E}[r|\, \mathbf{s}, \mathbf{a}]$. The mean reward of a quadratic reward domain given a state $\mathbf{s}$ and a action $\mathbf{a}$ is:

$$r(\mathbf{s}, \mathbf{a}) = -(\mathbf{s} - \mathbf{a})^T \begin{bmatrix} 11 & 9 \\ 9 & 11 \end{bmatrix} (\mathbf{s} - \mathbf{a}) \tag{51}$$

Dummy Action Dimension Domain Each dimension of actions and states are sampled uniform randomly in the range of $[-1, 1]$. The absolute error which is only dependent on the first dimension of action is designed as shown below.

$$r(\mathbf{s}, \mathbf{a}) = -\left|0.5 s_1 - a_1\right| \tag{52}$$

where the reward is not sampled from a distribution but determined by states $\mathbf{s}$ and actions $\mathbf{a}$.

Multi-Goal Reward Domain Each dimension of actions and states are sampled uniform randomly in the range of $[-1, 1]$. The multi-goal reward with exponential functions is designed as shown below.

$$\begin{aligned}
f_1(\mathbf{s}, \mathbf{a}) &= \exp\left(-\left(\left(\frac{(s_1 - a_1) - 0.5}{0.25}\right)^2 + \left(\frac{(s_2 - a_2)}{1}\right)^2\right)\right) \\
f_2(\mathbf{s}, \mathbf{a}) &= \exp\left(-\left(\left(\frac{(s_1 - a_1) + 0.5}{0.25}\right)^2 + \left(\frac{(s_2 - a_2)}{1}\right)^2\right)\right) \\
f_3(\mathbf{s}, \mathbf{a}) &= \exp\left(-\left(\left(\frac{(s_1 - a_1)}{1}\right)^2 + \left(\frac{(s_2 - a_2) + 0.5}{0.25}\right)^2\right)\right) \\
f_4(\mathbf{s}, \mathbf{a}) &= \exp\left(-\left(\left(\frac{(s_1 - a_1)}{1}\right)^2 + \left(\frac{(s_2 - a_2) - 0.5}{0.25}\right)^2\right)\right) \\
r(\mathbf{s}, \mathbf{a}) &= -\max\left(f_1(\mathbf{s}, \mathbf{a}), f_2(\mathbf{s}, \mathbf{a}), f_3(\mathbf{s}, \mathbf{a}), f_4(\mathbf{s}, \mathbf{a})\right)
\end{aligned} \tag{53}$$

where the reward is not sampled from a distribution but determined by states $\mathbf{s}$ and actions $\mathbf{a}$.

A.9.2 Warfarin Data Experiments

For the warfarin data, we used L2 regularizer for the 2 hidden layers with the coefficient of 1e-1. The L2 regularizer coefficient was chosen by conducting grid search over {1e-3, 1e-2, 1e-1}. We chose the coefficient that trains reward regressor in a stable manner and have low validation error for the reward regressor. Behavior policy density values below 1e-1. The clipping threshold is chosen by conducting grid search over {1e-3, 1e-2, 1e-1}. The clipping threshold that showed stable convergence and low MSE was chosen. The behavior policy density values are clipped for the same reason to the synthetic domain.

A.10 Effect of Error in a Reward Regression (Hessian Estimator for KMIS) on the Estimation Error and Convergence Speed of KMIS

Error Analysis To reduce the squared leading order bias $\mathrm{LOBIAS}(h, A(\mathbf{s}))^2$ in Eq. (54) (the equation can be derived with bias from Eq. (15), variance from Eq. (26), and the leading order bias with metric from Eq. (9)) of a kernel-relaxed IS estimator, KMIS learns a metric that reduces the absolute value of trace that is contained in the LOBIAS. To reduce the absolute value of the trace $|\operatorname{tr}(A(\mathbf{s})^{-1}H(\mathbf{s}))|$, we derived the optimal metric matrix $A(s)$ given the true Hessian matrix of expected reward w.r.t. an action selected by the target policy $H(\mathbf{s}) := \mathbf{H_a}\,\mathbb{E}[r|\,\mathbf{s},\mathbf{a}]|_{a=\pi(s)}$ in Appendix A.6. However, since we will be using estimated Hessian matrix $\widetilde{H}(\mathbf{s})$ acquired from a reward regressor, we analyze how the error in the reward regressor (error in the estimated Hessian $\widetilde{H}(\mathbf{s})$) affects KMIS estimation error.

$$\mathrm{MSE}(h, A(\mathbf{s}), N, D) = \underbrace{h^4 C_{b,A}}_{=:\mathrm{LOBIAS}(h,A(\mathbf{s}))^2} + O\left(h^6\right) + \frac{C_v}{Nh^D} + O\left(\frac{1}{Nh^{D-2}}\right) \tag{54}$$

where,

$$C_{b,A} := \left(\frac{1}{2}\mathbb{E}_{s\sim p(s)}\left[\operatorname{tr}\left(A(\mathbf{s})^{-1}H(\mathbf{s})\right)\right]\right)^2 \tag{55}$$

$$C_v := R(K)\mathbb{E}_{s\sim p(s)}\left[\frac{\mathbb{E}[r^2|s, a=\pi(s)]}{\pi_b(a=\pi(s)|s)}\right] \tag{56}$$

In Appendix A.6, we proved that when $H(\mathbf{s})$ contains both positive and negative eigenvalues, our metric makes the squared leading order bias $\mathrm{LOBIAS}(h, A(\mathbf{s}))^2 = 0$ by making the absolute value of the trace $|\operatorname{tr}(A(\mathbf{s})^{-1}H(\mathbf{s}))|$ zero. In this section we analyze the upper bound of $\mathrm{LOBIAS}(h, \widetilde{A}(\mathbf{s}))^2$ in the same setting. For analyzing, we use the following notations regarding the true Hessian $H(\mathbf{s})$, metric computed from the true Hessian $A(\mathbf{s})$, estimated Hessian from the reward regressor $\widetilde{H}(\mathbf{s})$ and the metric computed from it $\widetilde{A}(\mathbf{s})$:

$$H(\mathbf{s}) = [U_+(\mathbf{s})U_-(\mathbf{s})]\begin{pmatrix} \Lambda_+(\mathbf{s}) & 0 \\ 0 & \Lambda_-(\mathbf{s}) \end{pmatrix}[U_+(\mathbf{s})U_-(\mathbf{s})]^T \tag{57}$$

$$A(\mathbf{s}) = [U_+(\mathbf{s})U_-(\mathbf{s})]\begin{pmatrix} d_+(\mathbf{s})\Lambda_+(\mathbf{s}) & 0 \\ 0 & -d_-(\mathbf{s})\Lambda_-(\mathbf{s}) \end{pmatrix}[U_+(\mathbf{s})U_-(\mathbf{s})]^T \tag{58}$$

$$\Lambda(\mathbf{s}) = \begin{pmatrix} \Lambda_+(\mathbf{s}) & 0 \\ 0 & \Lambda_-(\mathbf{s}) \end{pmatrix}$$

$$\Lambda_+(\mathbf{s}) = \begin{pmatrix} \lambda_{+,1}(\mathbf{s}) & 0 & \cdots \\ 0 & \lambda_{+,2}(\mathbf{s}) & \\ \vdots & & \ddots \end{pmatrix}$$

$$U_+(\mathbf{s}) = \begin{pmatrix} | & | & \\ \mathbf{u}_{+,1}(\mathbf{s}) & \mathbf{u}_{+,2}(\mathbf{s}) & \cdots \\ | & | & \end{pmatrix}$$

$$\widetilde{H}(\mathbf{s}) = \left[\widetilde{U}_+(\mathbf{s})\widetilde{U}_-(\mathbf{s})\right]\begin{pmatrix} \widetilde{\Lambda}_+(\mathbf{s}) & 0 \\ 0 & \widetilde{\Lambda}_-(\mathbf{s}) \end{pmatrix}\left[\widetilde{U}_+(\mathbf{s})\widetilde{U}_-(\mathbf{s})\right]^T \tag{59}$$

$$\widetilde{A}(\mathbf{s}) = \left[\widetilde{U}_+(\mathbf{s})\widetilde{U}_-(\mathbf{s})\right]\begin{pmatrix} d_+(\mathbf{s})\widetilde{\Lambda}_+(\mathbf{s}) & 0 \\ 0 & -d_-(\mathbf{s})\widetilde{\Lambda}_-(\mathbf{s}) \end{pmatrix}\left[\widetilde{U}_+(\mathbf{s})\widetilde{U}_-(\mathbf{s})\right]^T \tag{60}$$

$$\widetilde{\Lambda}(\mathbf{s}) = \begin{pmatrix} \widetilde{\Lambda}_+(\mathbf{s}) & 0 \\ 0 & \widetilde{\Lambda}_-(\mathbf{s}) \end{pmatrix}$$

where we used eigendecomposition on $H(\mathbf{s})$. $\Lambda_+(\mathbf{s})$ and $\Lambda_-(\mathbf{s})$ are diagonal matrices containing positive and negative eigenvalues, respectively. $U_+(\mathbf{s})$ and $U_-(\mathbf{s})$ are matrices containing eigenvectors corresponding to $\Lambda_+(\mathbf{s})$ and $\Lambda_-(\mathbf{s})$, respectively. Similarly, $\widetilde{\Lambda}_+(\mathbf{s})$, $\widetilde{\Lambda}_-(\mathbf{s})$, $\widetilde{U}_+(\mathbf{s})$, $\widetilde{U}_-(\mathbf{s})$ are decomposed from $\widetilde{H}(\mathbf{s})$. We assume that $\widetilde{H}(\mathbf{s})$ correctly estimates the sign (positive, negative signs) of the eigenvalues in $H(\mathbf{s})$. Because of the assumption, there are d_+ and d_- in Eq. (60). We quantify the error in $\widetilde{H}(\mathbf{s})$ with constants ϵ and η ($\epsilon \geq 0$, $\eta > 0$) by:

$$
\begin{aligned}
|\widetilde{\mathbf{u}}_{a,i}(\mathbf{s})^T \mathbf{u}_{a,i}(\mathbf{s}) - 1| &\leq \epsilon && (61)\\
|\widetilde{\mathbf{u}}_{a,i}(\mathbf{s})^T \mathbf{u}_{b,j}(\mathbf{s})| &\leq \epsilon, \text{ if } a \neq b, \text{ or, } i \neq j && (62)\\
\|\widetilde{\Lambda}(\mathbf{s})^{-1}\Lambda(\mathbf{s}) - \mathbb{I}\| &\leq \epsilon && (63)\\
0 < \frac{\lambda_{a,i}(\mathbf{s})}{\widetilde{\lambda}_{a,j}(\mathbf{s})} &\leq \eta, \text{ if } i \neq j && (64)\\
-\eta \leq \frac{\lambda_{a,i}(\mathbf{s})}{\widetilde{\lambda}_{b,j}(\mathbf{s})} &< 0, \text{ if } a \neq b && (65)\\
&\text{where, } a, b \in \{+,-\}\\
&\quad\; i, j \in \{1,2,3,\cdots\}
\end{aligned}
$$

The trace term $\mathrm{tr}(\widetilde{A}(\mathbf{s})^{-1}H(\mathbf{s}))$ in the LOBIAS$(h, \widetilde{A}(\mathbf{s}))^2$ is,

$$
\begin{aligned}
&\mathrm{tr}(\widetilde{A}(\mathbf{s})^{-1}H(\mathbf{s}))\\
&= \mathrm{tr}\left([d_+\widetilde{U}_+\widetilde{\Lambda}_+\widetilde{U}_+^T - d_-\widetilde{U}_-\widetilde{\Lambda}_-\widetilde{U}_-^T]^{-1}[U_+\Lambda_+U_+^T + U_-\Lambda_-U_-^T]\right) && (66)\\
&= \mathrm{tr}\Bigg(\frac{1}{d_+}\widetilde{U}_+\widetilde{\Lambda}_+^{-1}\widetilde{U}_+^TU_+\Lambda_+U_+^T + \frac{1}{d_+}\widetilde{U}_+\widetilde{\Lambda}_+^{-1}\widetilde{U}_+^TU_-\Lambda_-U_-^T && (67)\\
&\qquad -\frac{1}{d_-}\widetilde{U}_-\widetilde{\Lambda}_-^{-1}\widetilde{U}_-^TU_+\Lambda_+U_+^T - \frac{1}{d_-}\widetilde{U}_-\widetilde{\Lambda}_-^{-1}\widetilde{U}_-^TU_-\Lambda_-U_-^T\Bigg)\\
&= \underbrace{\frac{1}{d_+}\left[\sum_{j=1}^{d_+}\sum_{i=1}^{d_+}\frac{\lambda_{+,i}(\mathbf{s})}{\widetilde{\lambda}_{+,j}(\mathbf{s})}(\widetilde{\mathbf{u}}_{+,j}(\mathbf{s})^T\mathbf{u}_{+,i}(\mathbf{s}))^2 + \sum_{j=1}^{d_+}\sum_{i=1}^{d_-}\frac{\lambda_{-,i}(\mathbf{s})}{\widetilde{\lambda}_{+,j}(\mathbf{s})}(\widetilde{\mathbf{u}}_{+,j}(\mathbf{s})^T\mathbf{u}_{-,i}(\mathbf{s}))^2\right]}_{=:X(\mathbf{s})} && (68)\\
&\quad - \underbrace{\frac{1}{d_-}\left[\sum_{j=1}^{d_-}\sum_{i=1}^{d_+}\frac{\lambda_{+,i}(\mathbf{s})}{\widetilde{\lambda}_{-,j}(\mathbf{s})}(\widetilde{\mathbf{u}}_{-,j}(\mathbf{s})^T\mathbf{u}_{+,i}(\mathbf{s}))^2 + \sum_{j=1}^{d_-}\sum_{i=1}^{d_-}\frac{\lambda_{-,i}(\mathbf{s})}{\widetilde{\lambda}_{-,j}(\mathbf{s})}(\widetilde{\mathbf{u}}_{-,j}(\mathbf{s})^T\mathbf{u}_{-,i}(\mathbf{s}))^2\right]}_{=:Y(\mathbf{s})}
\end{aligned}
$$

where in the third equality (Eq.68) we used,

$$
\mathrm{tr}(\widetilde{U}_a\widetilde{\Lambda}_a^{-1}\widetilde{U}_a^TU_b\Lambda_bU_b^T) = \sum_{j=1}^{d_a}\sum_{i=1}^{d_b}\frac{\lambda_{b,i}}{\widetilde{\lambda}_{a,j}}(\widetilde{\mathbf{u}}_{a,j}(\mathbf{s})^T\mathbf{u}_{b,j}(\mathbf{s}))^2, \text{where, } a, b \in \{+,-\} \tag{69}
$$

Expanding $X(\mathbf{s})$ we get,

$$
\begin{aligned}
X(\mathbf{s}) = \frac{1}{d_+}\Bigg[&\sum_{i=1}^{d_+}\frac{\lambda_{+,i}(\mathbf{s})}{\widetilde{\lambda}_{+,i}(\mathbf{s})}(\widetilde{\mathbf{u}}_{+,i}(\mathbf{s})^T\mathbf{u}_{+,i}(\mathbf{s}))^2 + \sum_{i\neq j}^{(d_+,d_+)}\frac{\lambda_{+,i}(\mathbf{s})}{\widetilde{\lambda}_{+,j}(\mathbf{s})}(\widetilde{\mathbf{u}}_{+,j}(\mathbf{s})^T\mathbf{u}_{+,i}(\mathbf{s}))^2 && (70)\\
&+\sum_{j=1}^{d_+}\sum_{i=1}^{d_-}\frac{\lambda_{-,i}(\mathbf{s})}{\widetilde{\lambda}_{+,j}(\mathbf{s})}(\widetilde{\mathbf{u}}_{+,j}(\mathbf{s})^T\mathbf{u}_{-,i}(\mathbf{s}))^2\Bigg]
\end{aligned}
$$

We get the following inequality for $X(\mathbf{s})$ by applying Eq. (61-65):

$$\underbrace{(1-\epsilon)^3 - d_-\eta\epsilon^2}_{=:X_l} \leq X(\mathbf{s}) \leq \underbrace{(1+\epsilon)^3 + (d_+ - 1)\eta\epsilon^2}_{=:X_u} \tag{71}$$

Similarly, we get,

$$\underbrace{(1-\epsilon)^3 - d_+\eta\epsilon^2}_{=:Y_l} \leq Y(\mathbf{s}) \leq \underbrace{(1+\epsilon)^3 + (d_- - 1)\eta\epsilon^2}_{=:Y_u} \tag{72}$$

The trace term $\mathrm{tr}(\widetilde{A}(\mathbf{s})^{-1}H(\mathbf{s}))$ $(= X(\mathbf{s}) - Y(\mathbf{s}))$ satisfies the following inequality:

$$X_l - Y_u \leq \mathrm{tr}(\widetilde{A}(\mathbf{s})^{-1}H(\mathbf{s})) \leq X_u - Y_l \tag{73}$$

Then the absolute value of the trace term satisfies the following inequality:

$$|\,\mathrm{tr}(\widetilde{A}(\mathbf{s})^{-1}H(\mathbf{s}))| \leq \max\{|X_u - Y_l|, |X_l - Y_u|\} \tag{74}$$

where, $|X_u - Y_l|$, $|X_l - Y_u|$ are,

$$|X_u - Y_l| = 6\epsilon + 2\epsilon^3 + \eta\epsilon^2(2d_+ - 1) \tag{75}$$

$$|X_l - Y_u| = 6\epsilon + 2\epsilon^3 + \eta\epsilon^2(2d_- - 1), \quad \text{where, } d_+, d_- \geq 1, \epsilon \geq 0, \eta > 0 \tag{76}$$

By defining $d_{max} := \max\{d_+, d_-\}$, we can rewrite Eq. (74):

$$|\,\mathrm{tr}(\widetilde{A}(\mathbf{s})^{-1}H(\mathbf{s}))| \leq \eta\epsilon^2(2d_{max} - 1) + 6\epsilon + 2\epsilon^3 \tag{77}$$

Then we get the following inequality for $\mathrm{LOBIAS}(h, \widetilde{A}(\mathbf{s}))^2$:

$$\mathrm{LOBIAS}(h, \widetilde{A}(\mathbf{s}))^2 \leq \frac{h^4}{4}\left[\eta\epsilon^2(2d_{max} - 1) + 6\epsilon + 2\epsilon^3\right]^2 \tag{78}$$

When there is no error in the Hessian estimation (when $\epsilon = 0$) acquired from the reward regressor, then the $\mathrm{LOBIAS}(h, \widetilde{A}(\mathbf{s}))^2$ becomes zero in Eq. (78). But when there is an error in the Hessian estimation (when $\epsilon > 0$), the KMIS estimator may have $\mathrm{LOBIAS}(h, \widetilde{A}(\mathbf{s}))^2$ up to $\frac{h^4}{4}\left[\eta\epsilon^2(2d_{max} - 1) + 6\epsilon + 2\epsilon^3\right]^2$.

Convergence Speed Analysis We analyzed the convergence speed of KMIS applied kernel-relaxed IS OPE estimator in Appendix A.7 when both positive and negative eigenvalues exist in the provided true Hessian $H(\mathbf{s})$, and given the optimal bandwidth h^*. In this section we analyze the convergence speed of KMIS estimator with $\widetilde{H}(\mathbf{s})$ in the same setting.

We derived the upper bound of $\mathrm{LOBIAS}(h, \widetilde{A}(\mathbf{s}))^2$ in Eq. (78). By plugging in the derived upper bound instead of $\mathrm{LOBIAS}(h, A(\mathbf{s}))^2$ in Eq. (54), and also using the relation $h^* = O\left(\left(\frac{D}{N}\right)^{\frac{1}{D+4}}\right)$ from Eq. (4), we get:

$$\begin{aligned}
&\mathrm{MSE}(h^*, \widetilde{A}(\mathbf{s}), N, D) \\
&\quad = (h^*)^4 O(d_{max}^2\eta^2\epsilon^4) + O\left((h^*)^6\right) + \frac{C_v}{N(h^*)^D} + O\left(\frac{1}{N(h^*)^{D-2}}\right), && (79)\\
&\quad = \mathcal{O}\left(\left(\frac{D}{N}\right)^{\frac{4}{D+4}} d_{max}^2\eta^2\epsilon^4\right) + \mathcal{O}\left(\left(\frac{D}{N}\right)^{\frac{4}{D+4}} \frac{1}{D}\right) && (80)\\
&\quad \text{where, } N \gg D
\end{aligned}$$

When the estimated Hessian $\widetilde{H}(\mathbf{s})$ has no error (when $\epsilon = 0$), the convergence speed derived in Eq. (80) matches the convergence speed of KMIS estimator with a true Hessian derived in Appendix A.7. However, when there is an error in the estimated Hessian (when $\epsilon > 0$), KMIS estimator with the estimated Hessian has a slower convergence rate to a true policy value than the KMIS estimator with a true Hessian by the rate of $O(D^{\frac{1}{2}}d_{max})$ (MSE of a KMIS estimator with the estimated Hessian converges to zero slower by the rate of $O(Dd_{max}^2)$ compared to the KMIS with a true Hessian).

Appendix References

[1] Nathan Kallus and Angela Zhou. Policy evaluation and optimization with continuous treatments. In *International conference on artificial intelligence and statistics*, pages 1243–1251. PMLR, 2018.

[2] Yung-Kyun Noh, Masashi Sugiyama, Kee-Eung Kim, Frank Park, and Daniel D Lee. Generative local metric learning for kernel regression. *Advances in Neural Information Processing Systems*, 30, 2017.

[3] Yung-Kyun Noh, Byoung-Tak Zhang, and Daniel D Lee. Generative local metric learning for nearest neighbor classification. In *Advances in neural information processing systems*, pages 1822–1830, 2010.

[4] Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661. PMLR, 2016.

[5] Yi Su, Pavithra Srinath, and Akshay Krishnamurthy. Adaptive estimator selection for off-policy evaluation. In *International Conference on Machine Learning*, pages 9196–9205. PMLR, 2020.