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 which is denoted as f in Eq. (3). The LOMSE is composed of constants C_b and C_v that are related to bias and variance, respectively :

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

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

where, $\nabla_{\mathbf{a}}^2$ 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 the kernel-relaxed IS estimate, the bias of the IS estimate can be reduced, and therefore, the MSE is reduced. Next, we see how the metric learning affects the convergence speed of our estimator w.r.t. datasize N and the action dimension D. In the following Theorem 2, we show that our estimator is consistent and converges faster than the kernel-relaxed IS estimator without metric learning.

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

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 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 are the work of Kallus and Zhou [2] and SLOPE [15], 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 $\mathbf{H}_{\mathbf{a}}\, r(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$, 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 $\mathbf{H}_{\mathbf{a}}\, r(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$ of constant and zero, respectively, we test our algorithm on an exponential reward where it has Talyor expansion terms larger than the second order.

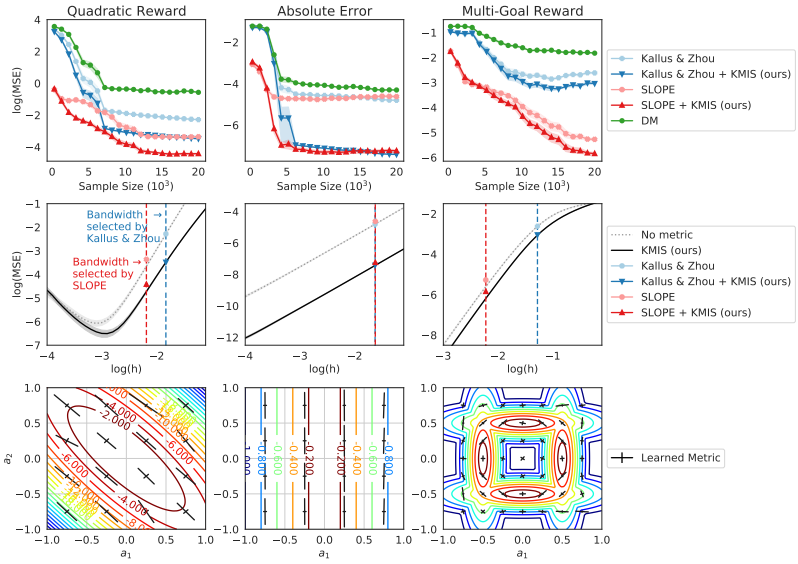


Figure 2: Experimental results on the synthetic data. Columns starting from the left are experimental results from the test domain of quadratic reward, negative absolute error with dummy action dimension, Multi-Goal reward made with the exponential functions. The first row shows how estimations are made 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. 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 plot empirical squared bias and variance of the Kallus&Zhou and Kallus&Zhou+KMIS estimates which together compose MSE on the absolute error domain with varying action dimensions 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.

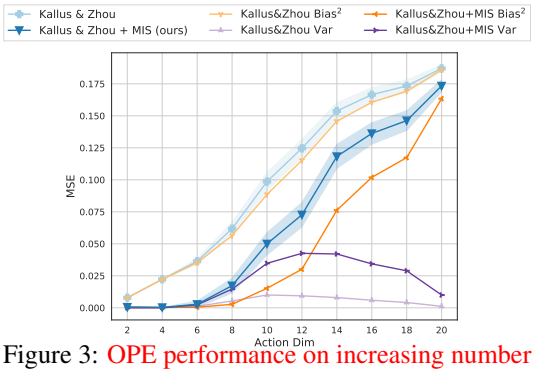
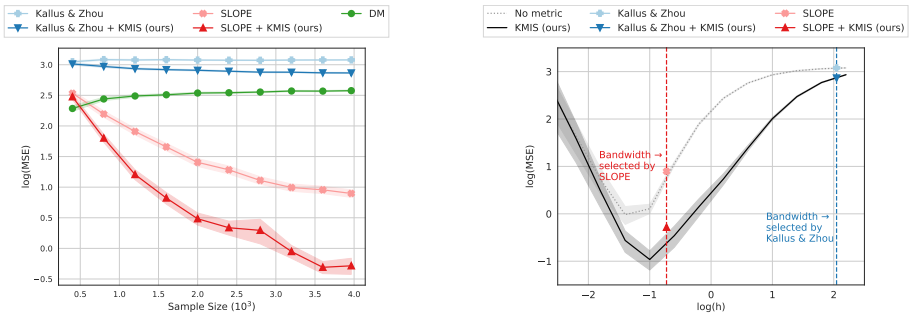


Figure 3: OPE performance on increasing number of action dimensions in the absolute error domain. Means and standard errors of squared errors obtained from 100 trials are drawn in the figures. Empirical bias and variance of the estimates are also drawn with the same estimates.

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]