Neural-Kernel Conditional Mean Embeddings

清水瑛貴 総合研究大学院大学 統計科学専攻 博士課程（5年一貫制）4 年

Kernel Conditional Mean Embeddings

Kernel conditional mean embeddings (CMEs) offer a powerful framework for representing conditional distributions. Let positive definite kernels $k_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ associated with RKHSs $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$, induce features $\psi(x) = k_{\mathcal{X}}(x, \cdot)$ and $\phi(y) = k_{\mathcal{Y}}(y, \cdot)$. The empirical estimate of CMEs (Song+, 2009) is expressed as:

$$\hat{\mu}_{P(Y|X)}(x) = \sum_{i=1}^{n} \beta_i(x)\phi(y_i) = \Phi\beta(x),$$

where:

$$\beta(x) = (K_X + \lambda I)^{-1} k_X.$$

Alternatively, this empirical estimate can be obtained by solving following function-valued regression problem (Grünewälder+, 2012):

$$\underset{C:\mathcal{H}_{\mathcal{X}} \to \mathcal{H}_{\mathcal{Y}}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \| \phi(y_i) - C\psi(x_i) \|_{\mathcal{H}_{\mathcal{Y}}}^2 + \lambda \|C\|_{HS}^2 .$$

3 challenges for CMEs

- **Scalability**: Gram matrix inverse can become prohibitively expensive for large dataset.
- **Expressiveness**: Pre-specified nature of RKHS features may lead to poor performance in practice.
- **Hyperparameter selection for $k_{\mathcal{Y}}$**: The objective function is defined in terms of the RKHS norm associated to $k_{\mathcal{Y}}$. Any change in kernel parameter fundamentally alters the objective.

Proposed Method

To address the scalability and expressiveness challenges, we propose the following form:

Neural Network (NN) Based CMEs

$$\hat{\mu}_{P(Y|X)}(x) = \sum_{a=1}^{M} \phi(\eta_a) f_a(x; \theta),$$

where $f(x;\theta) : \mathcal{X} \to \mathbb{R}^M$ represents a NN parameterized by θ, and $\eta_a \in \mathcal{Y}$ are M location parameters. The NN is optimized through $\min_\theta \frac{1}{n} \sum_{i=1}^{n} \hat{\ell}(\theta)$, where $\hat{\ell}(\theta)$ can be further expressed as:

$$\hat{\ell}(\theta) = -2 \sum_{a} k_{\mathcal{Y}}(y_i, \eta_a) f_a(x;\theta) + \sum_{a,b} k_{\mathcal{Y}}(\eta_a, \eta_b) f_a(x;\theta) f_b(x;\theta).$$

For $k_{\mathcal{Y}}$, we adopt following positive definite kernel $k_\sigma \in \mathcal{H}_\sigma$:

$$k_\sigma(y, y') = \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right)^{d_y} \exp\left(-\frac{\|y - y'\|^2}{2\sigma^2} \right).$$

Because this is also a smoothing kernel, we have $\hat{p}(y|x) = \langle k_\sigma(y, \cdot), \hat{\mu}_{P(Y|X)}(x) \rangle_{\mathcal{H}_\sigma} = \sum_{a=1}^{M} k_\sigma(y, \eta_a) f_a(x;\theta)$. We propose optimizing σ to minimize $\mathcal{L}_{SQ} = \frac{1}{2} \iint (\hat{p}(y|x) - p(y|x))^2 p(x) dx dy$, and obtain the empirical loss $\frac{1}{n} \sum_{i=1}^{n} \hat{\ell}_{SQ}(\sigma)$ where,

$$\hat{\ell}_{SQ}(\sigma) = -2 \sum_{a} k_\sigma(y_i, \eta_a) f_a(x;\theta) + \sum_{a,b} k_{\sqrt{2}\sigma}(\eta_a, \eta_b) f_a(x;\theta) f_b(x;\theta).$$

2 strategies for optimizing the hyperparameter σ

- **Iterative Optimization**: Optimize θ and σ, through $\min_\theta \frac{1}{n} \sum_{i=1}^{n} \hat{\ell}(\theta)$ and $\min_\sigma \frac{1}{n} \sum_{i=1}^{n} \hat{\ell}_{SQ}(\sigma)$, iteratively every step.
- **Joint Optimization**: $\min_{\theta,\sigma} \frac{1}{n} \sum_{i=1}^{n} \hat{\ell}(\theta, \sigma)$. This is justified from the fact that $\hat{\ell}_{SQ}(\sigma) \le \hat{\ell}(\sigma)$. Let $f = \sum_{a=1}^{M} k_{\sqrt{2}\sigma}(\cdot, \eta_a) w_a \in \mathcal{H}_{\sqrt{2}\sigma}$ and $g = \sum_{a=1}^{M} k_\sigma(\cdot, \eta_a) w_a \in \mathcal{H}_\sigma$. Then we can show that:

$$\|f\|_{\mathcal{H}_{\sqrt{2}\sigma}} \le \|g\|_{\mathcal{H}_\sigma}.$$

Experiments on Density Estimation Tasks

We evaluated on UCI datasets. For competitors:

- **Deep feature approach (DF)**: Replace ψ with a d-dimensional NN-parameterized feature map $\psi_\theta : \mathcal{X} \to \mathbb{R}^d$. For bandwidth selection, used the median heuristic or fixed to 0.1.
- **Diffusion based model (CARD)**: Combines a denoising diffusion-based conditional generative model with a pre-trained conditional mean estimator (Han+, 2022).

For evaluation mertic, we adopt QICE. We divide the generated samples into L quantile intervals with boundaries $\hat{y}_i^{\text{low}_j}$ and $\hat{y}_i^{\text{high}_j}$. Then compute QICE $:= \frac{1}{L} \sum_{j=1}^{L} \left| r_j - \frac{1}{L} \right|$, where $r_j = \frac{1}{n} \sum_{i=1}^{n} 1_{y_i \ge \hat{y}_i^{\text{low}_j}} \cdot 1_{y_i \le \hat{y}_i^{\text{high}_j}}$.

Dataset	QICE ↓				
	Iterative	Joint	DF-med	DF-0.1	CARD
Kin8nm	0.98 ± 0.29	$\mathbf{0.91 \pm 0.19}$	4.70 ± 0.32	2.60 ± 0.41	0.92 ± 0.25
Power	0.88 ± 0.24	$\mathbf{0.84 \pm 0.18}$	4.81 ± 0.33	2.91 ± 0.18	0.92 ± 0.21
Protein	$\mathbf{0.48 \pm 0.05}$	0.55 ± 0.17	1.83 ± 0.19	0.80 ± 0.07	0.71 ± 0.11

Applications to Reinforcement Learning

MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, R, P, \gamma)$, We represent the discounted sum of rewards received by an agent under a policy π as a random variable $Z^\pi(s,a) = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$ on space $\mathcal{Z}$. We propose to represent the distribution of $Z(s,a)$ using a kernel $k_{\mathcal{Z}}(\cdot, z) \in \mathcal{H}_{\mathcal{Z}}$:

$$\mu_{P(Z|S,A)}(x) = \sum_{a=1}^{M} k_{\mathcal{Z}}(\cdot, \eta_a) f_a(x;\theta),$$

where x is a tuple of state and action.

To construct a loss function in the context of deep Q-learning, we define a metric between the distribution of $R(s,a) + \gamma Z(s', a')$ and that of $Z(s,a)$. Maximum Mean Discrepancy provides a principled way to measure the discrepancy between these distributions:

Objective for Distributional Reinforcement Learning

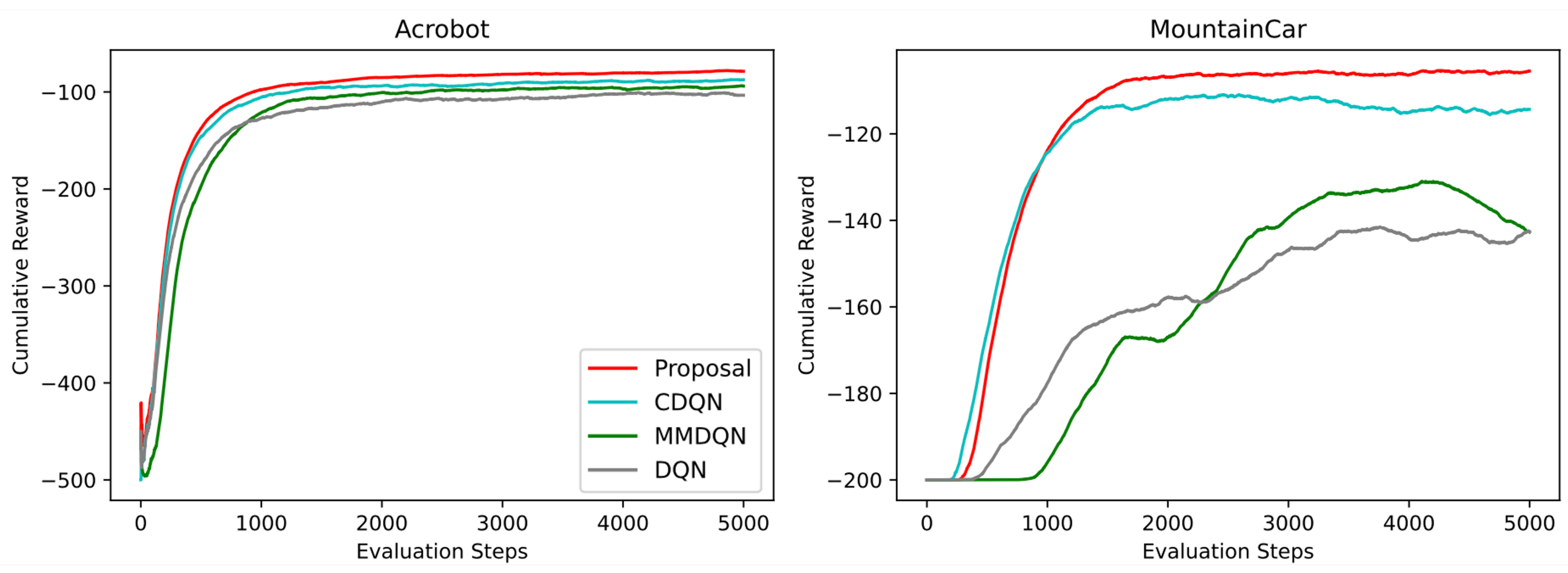
$$\hat{d}_{k_{\mathcal{Z}}} = \left\| \sum_{a=1}^{M} k_{\mathcal{Z}}(\cdot, \tau_a) v_a - \sum_{a=1}^{M} k_{\mathcal{Z}}(\cdot, \eta_a) w_a \right\|_{\mathcal{H}_{\mathcal{Z}}},$$

where $\tau_a = r + \gamma\eta_a$ and $v_a = f_a(x;\theta^-)$ which corresponds to the target network.

Inspired by MMD-FUSE (Biggs+, 2023) a two-sample testing method, we propose using a distribution over kernels, $k \in \mathcal{K}$, and employ the following log-sum-exp type loss function:

$$\text{FUSE} = \log\left(\mathbb{E}_{k\sim\omega}\left[\exp(\hat{d}_{k_{\mathcal{Z}}}^2(\theta)) \right] \right),$$

ω is an element of $\mathcal{M}(\mathcal{K})$, the set of distributions over the kernels.



Joint work with Kenji Fukumizu and Dino Sejdinovic