Fast Instrument Learning with Faster Rates

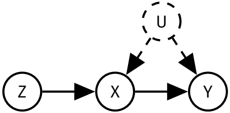
Ziyu Wang, Yuhao Zhou, Jun Zhu

Tsinghua University

find $f_0 : \mathcal{X} \to \mathrm{R}$ s.t.

$$\mathbb{E}(f_0(x) - y \mid z) = 0 \quad a.s.$$

x – treatment, y – outcome; z – *instruments*



Causal inference with confounded data: $\mathbb{E}(y \mid x = \cdot) \neq f_0$

Wright (1928); Newey and Powell (2003)

NPIV Estimation

Minimax estimator:

$$\hat{f}_n := \arg\min_{f \in \mathcal{H}} \max_{g \in \mathcal{I}} \frac{1}{n} \sum_{i=1}^{n} 2(f(x_i) - y_i - g(z_i))g(z_i) - \nu_n \|g\|_{\mathcal{I}}^2 + \lambda_n \|f\|_{\mathcal{H}}^2.$$

Inner loop: estimates the average violation of

$$\mathbb{E}(\mathbb{E}(f(x) - y \mid z)^2) = \|E(f - f_0)\|_{L_2(P_z)}^2$$

Cannot have a closed form unless $\mathcal{I}$ is an RKHS.

- $\Rightarrow$ Cannot do uncertainty quantification or model selection

Cannot prescribe a **good RKHS** $\mathcal{I}$ given high-dim z.

- Even though we only care about certain informative latent instruments

Dikkala et al (2020); Liao et al (2020); Muandet et al (2020)

Optimal Instrument Kernel for an RKHS $\mathcal{H}$

When x has moderate dimensions, and we have an RKHS $\mathcal{H}$ / kernel k_x –

There exists an "optimal instrument kernel" $(k_z, \mathcal{I})$ s.t.

$$f \sim \mathcal{GP}(0, k_x) \;\;\Rightarrow\;\; \mathbb{E}(f(\mathrm{x}) \mid \mathrm{z} = \cdot) \sim [\mathcal{GP}(0, k_z)]_{\sim}$$

But k_z involves E and must be *learned* from data

We can draw samples from $\mathcal{GP}(0, k_x)$, approximate $\mathbb{E}(f(\mathrm{x}) \mid \mathrm{z} = \cdot)$ with a regression oracle, and get "noisy samples" from $\mathcal{GP}(0, k_z)$

From Noisy GP Samples to a Learned Kernel

And given noisy samples from $\mathcal{GP}(0, k_z)$, we can efficiently learn k_z

Algorithm: implicitly construct $\tilde{k}_z \approx k_z$

1. Estimate $\hat{g}_{j,n_1} \leftarrow \mathrm{Regress}(\{(\tilde{z}_i, g_j(\tilde{z}_i) + \tilde{e}_{ij})\}_{i=1}^{n_1}) \approx g_j$ (noisy GP samples)
2. Return $k_z(z, z') := \frac{1}{2m} \sum_{j=1}^{2m} \hat{g}_{j,n_1}(z) \hat{g}_{j,n_1}(z')$,

Theorem ("test-time" approximation). Suppose the oracle satisfies

$$\mathbb{E}_{g \sim \mathcal{GP}(0,k_z)} \mathbb{E}_{D_1^{(n_1)}} \| g - \hat{g}_{n_1} \|_2^2 =: \xi_{n_1}^2.$$

Then, for $m \gtrsim m_0 \ll n^{1/(\bar{b}+1)}$, on an $D_1^{(n_1)}$-measurable event w.p.a. 1, we can have,

$$\text{for any } g_* \in \mathcal{I}_0, \exists \tilde{g} \in \mathcal{I} \text{ s.t. } \| \tilde{g} - g_* \|_2 = \tilde{O}\left(\xi_{n_1} + n_1^{-\bar{b}/2(\bar{b}+1)}\right).$$

(Briefly, we don't need to worry about the complexity of $\mathcal{I}$.)

Idea: we learned enough about the leading Mercer eigenfunctions.

From Noisy GP Samples to a Learned Kernel

And given noisy samples from $\mathcal{GP}(0, k_z)$, we can efficiently learn k_z

Algorithm: implicitly construct $\tilde{k}_z \approx k_z$

1. Estimate $\hat{g}_{j,n_1} \leftarrow \mathrm{Regress}(\{(\tilde{z}_i, g_j(\tilde{z}_i) + \tilde{e}_{ij})\}_{i=1}^{n_1}) \approx g_j$ (noisy GP samples)
2. Return $k_z(z, z') := \frac{1}{2m} \sum_{j=1}^{2m} \hat{g}_{j,n_1}(z) \hat{g}_{j,n_1}(z')$,

Theorem ("test-time" approximation). Suppose the oracle satisfies

$$\mathbb{E}_{g \sim \mathcal{GP}(0,k_z)} \mathbb{E}_{D_1^{(n_1)}} \|g - \hat{g}_{n_1}\|_2^2 =: \xi_{n_1}^2.$$

Then, for $m \gtrsim m_0 \ll n^{1/(\bar{b}+1)}$, on an $D_1^{(n_1)}$-measurable event w.p.a. 1, we can have,

$$\text{for any } g_* \in \mathcal{I}_0, \exists \tilde{g} \in \mathcal{I} \text{ s.t. } \|\tilde{g} - g_*\|_2 = \tilde{O}\left(\xi_{n_1} + n_1^{-\bar{b}/2(\bar{b}+1)}\right).$$

(Briefly, we don't need to worry about the complexity of $\mathcal{I}$.)

Also models **multi-task learning**, where $\{g_j\}$ / g_* are training / test-time tasks[1]

[1] Improves over (Tripuraneni et al, 2020; Du et al, 2021) for GP models

NPIV Estimation Results Illustrated

A concrete high-dimensional example:

- The feature extractor $\Phi_0 \in \mathcal{C}^{\beta_1}(\mathbb{R}^{d_1} \to \mathbb{R}^{d_2})$, where $\frac{d_1}{d_2} = \frac{\dim z}{\dim \bar{z}} \gg 1$, and $\frac{\beta_1}{d_1} \gtrsim 1$.[2]
- True latent-space $\bar{k}_z^0$ is equivalent to Matérn-β_2, where $\bar{b} = 2\beta_2/d_2 \gtrsim 1$.

choice for $\mathcal{I}$	fixed-form	learned	k_z (unusable)
Polynomial rate	$\frac{\beta_1}{2\beta_1+d_1} \wedge \frac{\bar{b}}{2(\bar{b}+d_1/d_2)}$	$\left(\frac{\beta_1}{2\beta_1+d_1} \wedge \frac{\bar{b}}{2(\bar{b}+3)}\right)\frac{\bar{b}}{\bar{b}+1}$	$\frac{\bar{b}}{2(\bar{b}+1)}$

Table: Convergence rates for $\|\hat{f}_n - f_0\|_2$ w.r.t. $n = n_1 + n_2$.

Learned kernel avoids the curse of dimensionality.

[2]To simplify notations; we can compare the other side as well.

Setup: optionally extends Bennett et al (2019) with high-dim instruments

Baselines: fixed-form kernels (-RBF), flexible models (-Tree, -NN)

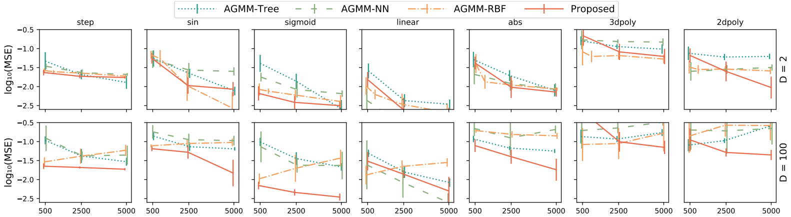


Figure: Test MSE across all settings.

Method	AGMM-Tree	AGMM-NN	**Proposed**
Runtime / s	1374 ± 418	303 ± 16	**25.9 ± 5.6**

Simulation: Uncertainty Quantification

Method	$n_1 = n_2$	Test MSE	90% CB. Rad.	90% CB. Cvg.	90% CI. Cvg.
	500	.431 ±.192	.240 ±.036	.187 [.147, .235]	.640 ±.191
RBF $\mathcal{I}$	2500	.176 ±.089	.175 ±.023	.517 [.460, .573]	.822 ±.136
	5000	.126 ±.072	.156 ±.019	.660 [.605, .711]	.855 ±.143
	500	.097 ±.065	.201 ±.025	.923 [.888, .948]	.915 ±.123
Proposed	2500	.035 ±.024	.074 ±.008	.917 [.880, .943]	.908 ±.127
	5000	.024 ±.016	.049 ±.004	.920 [.884, .946]	.905 ±.134

Table: Test MSE, radius and coverage rate of the 90% L_2 credible ball (CB) / pointwise CI, for $f_0 \sim \mathcal{GP}$, $D = 100$.

Learned $\mathcal{I}$ leads to **reliable** credible sets which are **also more informative**.

Idea: learns optimal[3] tensor product kernels for $\mathcal{H}$ and $\mathcal{I}$

Experiment on the demand data (Hartford et al, 2017)

n	DeepIV	DeepGMM	AGMM-RBF	AGMM-NN	**Proposed**
Low-dimensional setting					
1000	3.76 [3.74, 3.77]	3.97 [3.94, 3.99]	3.75 [3.71, 3.79]	3.42 [3.06, 3.99]	**2.94 [2.85, 3.06]**
5000	3.14 [3.10, 3.21]	3.94 [3.91, 3.96]	3.50 [3.46, 3.52]	2.74 [2.66, 2.76]	**2.39 [2.30, 2.47]**
Image setting					
5000	3.96 [3.93, 4.01]	4.41 [4.38, 4.45]	4.03 [4.02, 4.05]	4.20 [4.10, 4.33]	**3.87 [3.85, 3.92]**

Table: Log test MSE vs the total sample size ($n = n_1 + n_2$).

[3] In the sense of minimizing $\|E(\hat{f}_n - f_0)\|_2$; no estimation theory established

Thanks!

Paper: `https://arxiv.org/abs/2205.10772`

Code: `https://github.com/meta-inf/fil`