Learning conditionally independent representations with kernel regularizers

Roman Pogodin*
Gatsby → Mila

Namrata Deka*
UBC → CMU

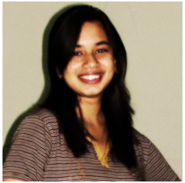
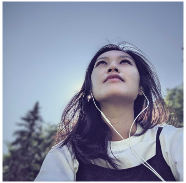
Yazhe Li*
Gatsby + DeepMind

Danica J. Sutherland
UBC + Amii

Victor Veitch
UChicago + Google

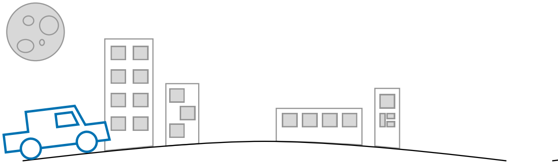
Arthur Gretton
Gatsby

Gatsby25, June 2023
based on arXiv:2212.08645 (ICLR 2023, “notable: top 5%”)

Intro: conditionally invariant representations

- Self-driving car tries to predict its location
- Starts in the morning
- Finishes in the evening
- . . . learns to predict **location** from **time of day**



Distribution shift: car starts in the afternoon

- ...and makes lots of errors

Intro: conditionally invariant representations

Idealized solution to this **distribution shift** problem:

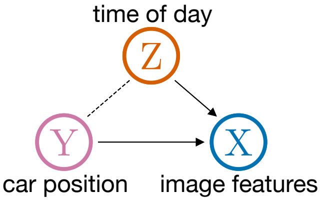
- predictions should be **conditionally independent** of time given the car position: $X \perp\!\!\!\perp Z \mid Y$

Same form as a common **domain invariance** objective:

features $\perp\!\!\!\perp$ domain ID $\mid$ true label

Same form as common **fairness** criterion (equalized odds):

predictions $\perp\!\!\!\perp$ protected attribute $\mid$ true label



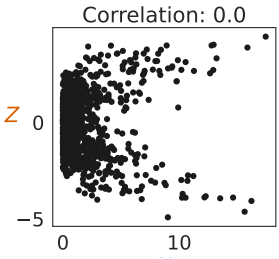
Problem: *conditional* dependence is **hard to measure**!

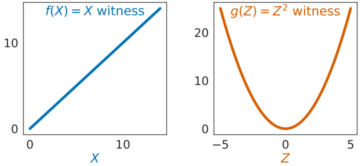
- Discrete Y: check dependence of X and Z for *each* Y value
  - On *each minibatch* during training. . .
- Continuous Y: prior work runs regression on each minibatch

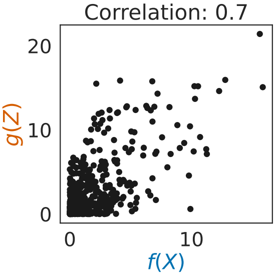
Warmup: detecting **unconditional** dependence

$$\begin{aligned} Y &\sim \mathcal{N}(0,1) \\ \xi_1, \xi_2 &\sim \mathcal{N}(0,1) \text{ i.i.d. noise} \\ X &= (Y + \xi_1)^2 \\ Z &= Y + \xi_1 + \xi_2 \end{aligned}$$

- X and Z are **uncorrelated**





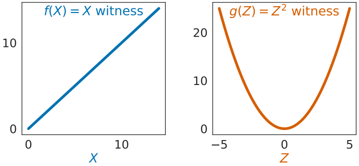


$X \perp\!\!\!\perp Z$ if and only if **all** square-integrable functions $f(X)$ and $g(Z)$ are uncorrelated

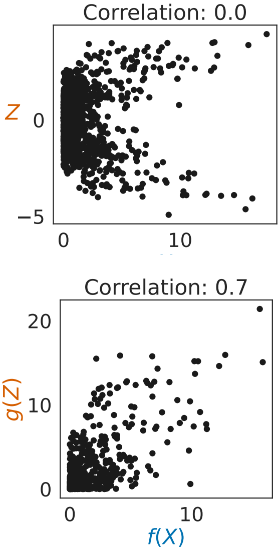
Warmup: detecting **unconditional** dependence

$$\begin{aligned} Y &\sim \mathcal{N}(0,1) \\ \xi_1, \xi_2 &\sim \mathcal{N}(0,1) \text{ i.i.d. noise} \\ X &= (Y + \xi_1)^2 \\ Z &= Y + \xi_1 + \xi_2 \end{aligned}$$

- X and Z are **uncorrelated**
- One way to detect dependence: we can find correlated **nonlinear** functions $f(X)$ and $g(Z)$

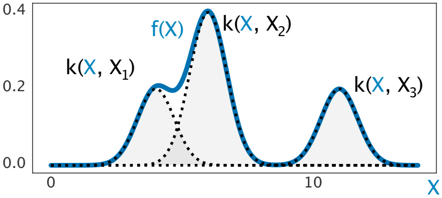


$X \perp\!\!\!\perp Z$ if and only if **all** square-integrable functions $f(X)$ and $g(Z)$ are uncorrelated



Warmup: detecting **unconditional** dependence

- If there aren't *any* correlated $f(X)$ and $g(Z)$, then X and Z are independent
- How to check ~~all~~ *enough* nonlinear functions?
- Check $f(X)$ and $g(Z)$ from **kernel spaces** (RKHSes): $f(X) = \sum_i \alpha_i \, k(X, X_i)$



- From RKHS properties: $\operatorname{Cov}\left(f(X), g(Z)\right) = \langle f, C_{XZ} \, g \rangle$ for the linear operator

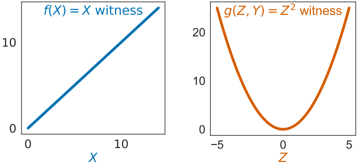
$$C_{XZ} = \mathbb{E}[k(X, \cdot) \otimes k(Z, \cdot)] - \mathbb{E}[k(X, \cdot)] \otimes \mathbb{E}[k(Z, \cdot)]$$

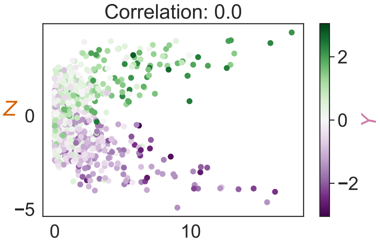
  - With linear kernels, C_{XZ} is just the cross-covariance matrix $\mathbb{E}[X Z^\top] - \mathbb{E}[X] \, \mathbb{E}[Z]^\top$
  - If $C_{XZ} = 0$, all $f(X)$ and $g(Z)$ in the RKHSes are uncorrelated
  - If our kernels are "rich enough" (Gaussian is enough), this implies independence
- Hilbert-Schmidt Independence Criterion: $\operatorname{HSIC}(X, Z) = \|C_{XZ}\|_{\mathrm{HS}}^2 = 0$ iff $C_{XZ} = 0$
  - Can estimate with $\widehat{\operatorname{HSIC}}(X, Z) = \frac{1}{B^2} \mathbf{1}^\top \left(H K_{XX} H \odot K_{ZZ} \right) \mathbf{1}$, where H is "centring matrix"
- Deep nets with features X_θ ~independent of Z: $\min_\phi \operatorname{loss}(\phi(X), Y) + \gamma \, \widehat{\operatorname{HSIC}}(\phi(X), Z)$

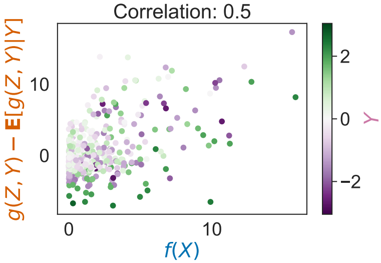
Detecting **conditional** dependence

$$\begin{aligned} Y &\sim \mathcal{N}(0,1) \\ \xi_1, \xi_2 &\sim \mathcal{N}(0,1) \text{ i.i.d. noise} \\ X &= (Y + \xi_1)^2 \\ Z &= Y + \xi_1 + \xi_2 \end{aligned}$$

- X and Z are **dependent**
- X and Z are **conditionally dependent** given Y (through ξ_1)







$X \perp\!\!\!\perp Z \mid Y$ if and only if **all** $f(X)$ are uncorrelated with **all** $g(Z,Y) - \mathbb{E}\left[g(Z,Y) \mid Y\right]$ [Daudin, 1980]

CIRCE: Conditional Independence Regression CovariancE

- Want to check covariance of $f(X)$ and $g^c(Z,Y) = g(Z,Y) - \mathbb{E}\left[g(Z,Y) \mid Y\right]$
  - $g^c(Z,Y)$ has mean zero, so they're uncorrelated iff $\mathbb{E}[f(X)\, g^c(Z,Y)] = 0$
- The **CIRCE operator** gives $\langle f, C^c_{XZ|Y} g\rangle = \mathbb{E}[f(X)\, g^c(Z,Y)]$, using

$$C^c_{XZ|Y} = \mathbb{E}\Big[k(X,\cdot) \otimes \big(k((Z,Y),\cdot) - \mathbb{E}[k((Z',Y),\cdot) \mid Y]\big)\Big]$$

- $\mathrm{CIRCE}(X, Z \mid Y) = \|C^c_{XZ|Y}\|^2_{\mathrm{HS}} = 0$ iff $X \perp\!\!\!\perp Z \mid Y$, if kernels are "rich enough"

- Special case: if $k((Z,Y),(Z',Y')) = k(Z,Z')\, k(Y,Y')$, we get

$$C^c_{XZ|Y} = \mathbb{E}[k(X,\cdot) \otimes k(Y,\cdot) \otimes \big(k(Z,\cdot) - \mu_{Z|Y}(Y)\big)]$$

where $\mu_{Z|Y}$ is the **conditional mean embedding** of Z given Y

CIRCE estimator

- Want squared norm of $C^c_{XZ|Y} = \mathbb{E}[k(X,\cdot) \otimes k(Y,\cdot) \otimes \big(k(Z,\cdot) - \mu_{Z|Y}(Y)\big)]$
- First, estimate conditional mean embedding $\mu_{Z|Y}$ on a dataset $\{(Z_i, Y_i)\}_{i=1}^M$
  - Use kernel ridge regression: inputs Y, RKHS-valued labels $k(Z,\cdot)$
  - Use this to estimate the conditionally-centred kernel function

$$\begin{aligned}\hat{k}^c((Z,Y),(Z',Y')) &= \langle k(Z,\cdot) - \hat{\mu}_{Z|Y}(Y), k(Z',\cdot) - \hat{\mu}_{Z|Y}(Y')\rangle \\ &\approx k(Z,Z') - \mathbb{E}[k(Z,Z') \mid Y] - \mathbb{E}[k(Z,Z') \mid Y'] + \mathbb{E}[k(Z,Z') \mid Y, Y']\end{aligned}$$

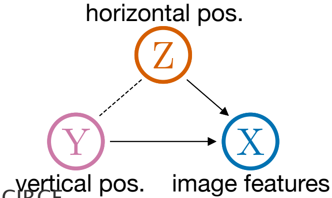
- While training $\phi(X)$, for each batch $\{(\phi(X_i), Z_i, Y_i)\}_{i=1}^B$:
  - Get $(K_{XX})_{ij} = k(\phi(X_i), \phi(X_j))$, $(K_{YY})_{ij} = k(Y_i, Y_j)$, $(\hat{K}^c_{ZZ})_{ij} = \hat{k}^c((Z,Y),(Z',Y'))$
  - Regularize with $\widehat{\mathrm{CIRCE}} = \frac{1}{B(B-1)} \mathbf{1}^\top \Big(K_{XX} \odot K_{YY} \odot \hat{K}^c_{ZZ}\Big)\mathbf{1}$

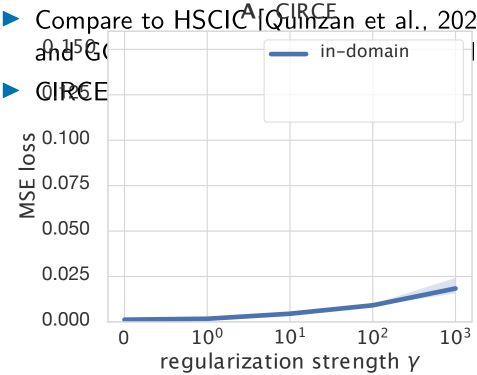
Benefits of CIRCE:

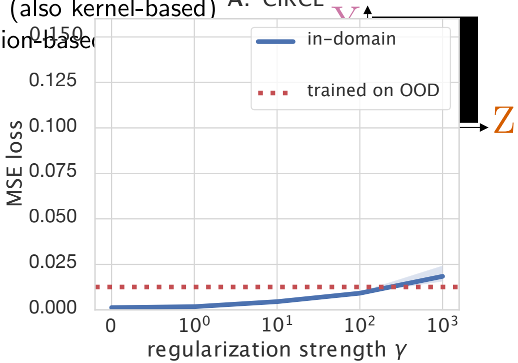
- As $B, M \to \infty$, $\widehat{\mathrm{CIRCE}} \to 0$ iff $\phi(X) \perp\!\!\!\perp Z \mid Y$; rate is known (see paper)
- K_{YY} and $\hat{K}^c_{ZZ}$ don't depend on ϕ:
  - Can precompute them, so only need $k(\phi(X_i), \phi(X_j))$ for each new ϕ
  - Separates (small) batch size B and (big) regression training size M: better convergence

Experiments

- dSprites dataset [Matthey et al., 2017]:
  2D shapes in different locations
- Task: predict vertical position Y
  But be invariant to horizontal position Z
  Z and Y have strong dependence in training
- Compare to HSCIC [Quinzan et al., 2022] (also kernel-based)
  and G lation-based
- CIRCE

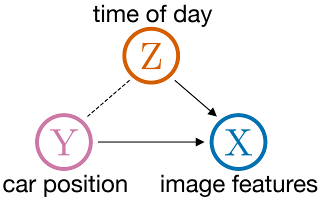






A. CIRCE

Discussion



- **CIRCE**: a measure of conditional independence for feature learning
- It works with continuous variables and in deep learning settings
- Applications: domain shift invariance, fairness

arXiv paper

(code link inside)