Learning conditionally independent representations with kernel regularizers

Roman Pogodin*  Namrata Deka*  Yazhe Li*
Gatsby → Mila  UBC → CMU  Gatsby + DeepMind

Danica J. Sutherland  Victor Veitch  Arthur Gretton
UBC + Amii  UChicago + Google  Gatsby

Gatsby25, June 2023
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

\[ Y \sim \mathcal{N}(0, 1) \]
\[ \xi_1, \xi_2 \sim \mathcal{N}(0, 1) \text{ i.i.d. noise} \]
\[ X = (Y + \xi_1)^2 \]
\[ Z = Y + \xi_1 + \xi_2 \]

\[ \rightarrow \] \( X \) and \( Z \) are **uncorrelated**

\[ X \perp Z \text{ if and only if all square-integrable functions } f(X) \text{ and } g(Z) \text{ are uncorrelated} \]
Warmup: detecting unconditional dependence

\[ Y \sim \mathcal{N}(0, 1) \]
\[ \xi_1, \xi_2 \sim \mathcal{N}(0, 1) \text{ i.i.d. noise} \]
\[ X = (Y + \xi_1)^2 \]
\[ Z = Y + \xi_1 + \xi_2 \]

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

\[ f(X) = X \text{ witness} \]
\[ g(Z) = Z^2 \text{ witness} \]

\[ X \perp \!\!\!\!\perp Z \text{ if and only if all square-integrable functions } f(X) \text{ and } g(Z) \text{ are uncorrelated} \]
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: $\text{Cov}(f(X), g(Z)) = \langle f, C_{XZ} g \rangle$ 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 $\mathbb{E}[XZ^\top] - \mathbb{E}[X] \mathbb{E}[Z]^\top$
  - 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: $\text{HSIC}(X, Z) = \|C_{XZ}\|_{\text{HS}}^2 = 0$ iff $C_{XZ} = 0$
  - Can estimate with $\hat{\text{HSIC}}(X, Z) = \frac{1}{B^2} 1^\top (HK_{XX} H \otimes K_{ZZ}) 1$, where $H$ is “centring matrix”
- Deep nets with features $X_\theta \sim$ independent of $Z$: $\min_{\phi} \text{loss}(\phi(X), Y) + \gamma \hat{\text{HSIC}}(\phi(X), Z)$
Detecting **conditional** dependence

\[ Y \sim \mathcal{N}(0, 1) \]
\[ \xi_1, \xi_2 \sim \mathcal{N}(0, 1) \text{ i.i.d. noise} \]
\[ X = (Y + \xi_1)^2 \]
\[ Z = Y + \xi_1 + \xi_2 \]

- \( X \) and \( Z \) are **dependent**
- \( X \) and \( Z \) are **conditionally dependent** given \( Y \) (through \( \xi_1 \))

\[ X \perp Z \mid Y \text{ if and only if all } f(X) \text{ are uncorrelated with all } g(Z, Y) - \mathbb{E}[g(Z, Y) \mid Y] \] [Daudin, 1980]
CIRCE: Conditional Independence Regression CovariancE

- Want to check covariance of $f(X)$ and $g^c(Z, Y) = g(Z, Y) - \mathbb{E}[g(Z, Y) | Y]$
  - $g^c(Z, Y)$ 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

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

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

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

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

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

- Want squared norm of \( C_{XZ|Y}^c = \mathbb{E}[k(X, \cdot) \otimes k(Y, \cdot) \otimes (k(Z, \cdot) - \mu_{Z|Y}(Y))] \)

- 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

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

- While training \( \phi(X) \), for each batch \( \{((\phi(X_i), Z_i, Y_i))\}_{i=1}^B \):
  - Get \( (K_{XX})_{ij} = k(\phi(X_i), \phi(X_j)), (K_{YY})_{ij} = k(Y_i, Y_j), (\hat{K}_{ZZ}^c)_{ij} = \hat{k}^c((Z, Y), (Z', Y')) \)
  - Regularize with \( \underset{\text{CIRCE}}{\text{CIRCE}} = \frac{1}{B(B-1)} \mathbf{1}^T \left( K_{XX} \otimes K_{YY} \otimes \hat{K}_{ZZ}^c \right) \mathbf{1} \)

Benefits of CIRCE:

- As \( B, M \to \infty \), \( \text{CIRCE} \to 0 \) iff \( \phi(X) \perp \perp Z | Y \); rate is known (see paper)

- \( K_{YY} \) and \( \hat{K}_{ZZ}^c \) don’t depend on \( \phi \):
  - Can precompute them, so only need \( k(\phi(X_i), \phi(X_j)) \) for each new \( \phi \)
  - 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 GCM [Shah & Peters, 2020] (correlation-based)
- CIRCE wins!

Diagram

- $X$: horizontal pos.
- $Y$: vertical pos.
- $Z$: image features

MSE loss vs. regularization strength $\gamma$

- **A. CIRCE**
  - Blue line: in-domain
  - Dotted red line: trained on OOD

- **B. HSCIC**
  - Same as A but for HSCIC

- **C. GCM**
  - Same as A but for GCM
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

![Diagram](image-url)