東京大学 THE UNIVERSITY OF TOKYO

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

RIKEN

Generalizing Importance Weighting to A Universal Solver for Distribution Shift Problems

Tongtong Fang[1]

Nan Lu[2,3]

Gang Niu[3]

Masashi Sugiyama[3,1]

[1] Univ. of Tokyo, Japan,

[2] Univ. of Tübingen, Germany,

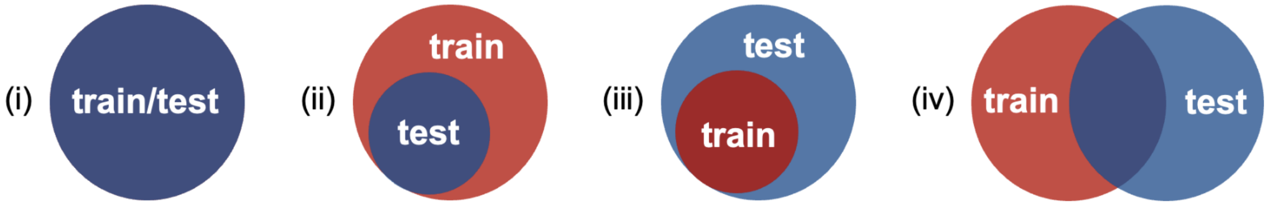
[3] RIKEN, Japan

Two-levels of distribution shift

- Two levels of distribution shift (DS) $p_{\mathrm{tr}}(\boldsymbol{x}, y) \neq p_{\mathrm{te}}(\boldsymbol{x}, y)$:
  - The data distribution itself changes, e.g., covariate shift
  - The underlying *support** of data distribution changes.

** The set where the probability density is non-zero.*

- The relationship between the training and test support



- Existing methods are good at cases (i) & (ii)
- Cases (iii) & (iv) are common due to data-collection biases

A real-world example of case (iii)

Family Felidae (猫科, ネコ科)

Training data:

Tiger, ~~ocelot~~, cat, ~~leopard~~, ~~puma, caracal, lion…~~

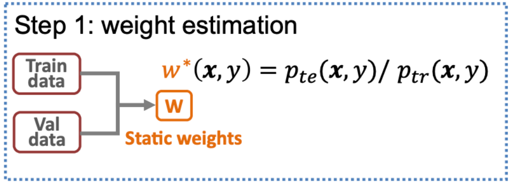
Test data:

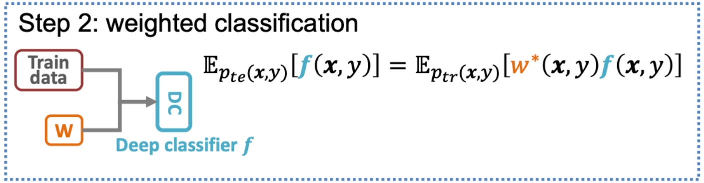
Tiger, ocelot, cat, leopard, puma, caracal, lion…

Pictures are from Wikimedia Commons.

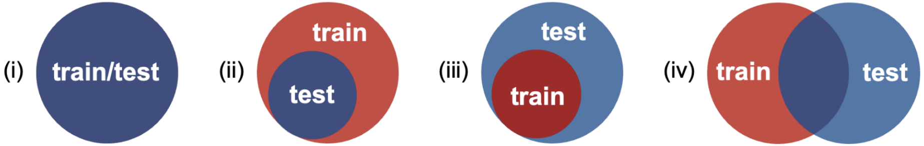
Importance weighting (IW) & dynamic IW

- Problem setting:
  - A training set $\{(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}} \sim p_{\mathrm{tr}}(\boldsymbol{x}, y)$ & a validation set $\{(\boldsymbol{x}_i^{\mathrm{v}}, y_i^{\mathrm{v}})\}_{i=1}^{n_{\mathrm{v}}} \sim p_{\mathrm{te}}(\boldsymbol{x}, y)$, $n_{\mathrm{tr}} \gg n_{\mathrm{v}}$
  - Estimate the risk: $R(\boldsymbol{f}) = \mathbb{E}_{p_{\mathrm{te}}(\boldsymbol{x},y)}[\ell(\boldsymbol{f}(\boldsymbol{x}), y)]$.
- IW is a golden solver for DS in cases (i) and (ii)[1]





- Dynamic IW (DIW) makes IW work well for deep learning[2]
- However, IW methods are problematic in cases (iii) & (iv)



[1] M. Sugiyama et al., Machine learning in non-stationary environments: Introduction to covariate shift adaptation. The MIT Press, 2012.
[2] T. Fang et al., Rethinking importance weighting for deep learning under distribution shift. In NeurIPS, 2020.

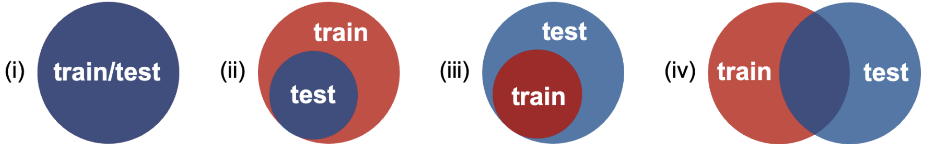
A deeper understanding of IW

- Recall that the risk is: $R(\boldsymbol{f}) = \mathbb{E}_{p_{\mathrm{te}}(\boldsymbol{x},y)}[\ell(\boldsymbol{f}(\boldsymbol{x}), y)]$
- The objective of IW is: $J(\boldsymbol{f}) = \mathbb{E}_{p_{\mathrm{tr}}(\boldsymbol{x},y)}[w^*(\boldsymbol{x}, y)\ell(\boldsymbol{f}(\boldsymbol{x}), y)]$

Definition: Given an (expected) objective $J(\boldsymbol{f})$, it is risk-consistent if $J(\boldsymbol{f}) = R(\boldsymbol{f})$ for any $\boldsymbol{f}$, i.e., the objective is equal to the original risk for any classifier; it is classifier-consistent if $\arg\min_{\boldsymbol{f}} J(\boldsymbol{f}) = \arg\min_{\boldsymbol{f}} R(\boldsymbol{f})$ where the minimization is taken over all measurable functions, i.e., the objective shares the optimal classifier with the original risk.

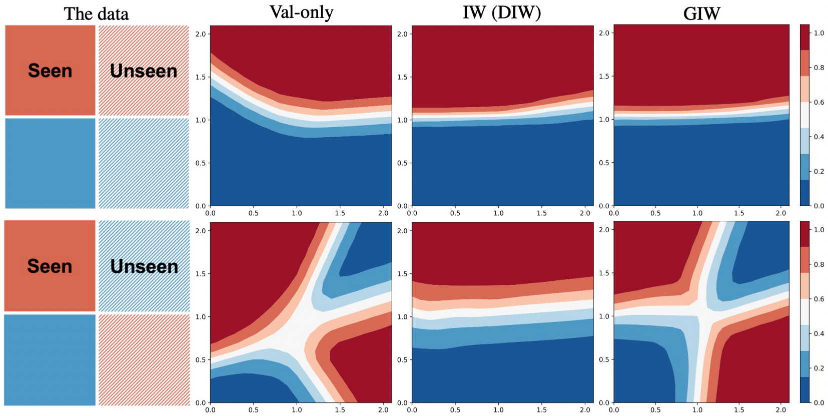
Theorem 1. *In cases (i) and (ii), IW is risk-consistent.*

Theorem 2. *In cases (iii) and (iv), IW is risk-inconsistent, and it holds that $J(\boldsymbol{f}) < R(\boldsymbol{f})$ for any $\boldsymbol{f}$.*



Two concrete examples

- **Task**: classify red/blue data
- **Data**:
  - Train: left two squares
  - Test: all four squares
  - Validation: 1 data per square
- **Methods**:
  - Val-only: use only validation data to train the model
  - IW: importance weighting
  - GIW: the proposed method



IW is classifier-consistent

IW is classifier-inconsistent

Generalized importance weighting (GIW)

- We split the test support $\mathcal{S}_{\mathrm{te}}$ into in-training $\mathcal{S}_{\mathrm{te}} \cap \mathcal{S}_{\mathrm{tr}}$ and out-of-training $\mathcal{S}_{\mathrm{te}} \backslash \mathcal{S}_{\mathrm{tr}}$ parts.
- A support-splitting variable $s \in \{0,1\}$, then

$$p(\boldsymbol{x}, y, s) = \begin{cases} p_{\mathrm{te}}(\boldsymbol{x}, y) & \text{if } (\boldsymbol{x}, y) \in \mathcal{S}_{\mathrm{tr}} \text{ and } s = 1, \text{or } (\boldsymbol{x}, y) \in \mathcal{S}_{\mathrm{te}} \backslash \mathcal{S}_{\mathrm{tr}} \text{ and } s = 0, \\ 0 & \text{if } (\boldsymbol{x}, y) \in \mathcal{S}_{\mathrm{tr}} \text{ and } s = 0, \text{or } (\boldsymbol{x}, y) \in \mathcal{S}_{\mathrm{te}} \backslash \mathcal{S}_{\mathrm{tr}} \text{ and } s = 1. \end{cases}$$

- Expected objective of GIW is:

$$J_G(\boldsymbol{f}) = \underbrace{\alpha \mathbb{E}_{p_{\mathrm{tr}}(\boldsymbol{x}, y)}[w^*(\boldsymbol{x}, y)\ell(f(\boldsymbol{x}), y)]}_{\text{in-training (IT)}} + \underbrace{(1-\alpha)\mathbb{E}_{p(\boldsymbol{x}, y \mid s = 0)}[\ell(f(\boldsymbol{x}), y)]}_{\text{out-of-training (OOT)}},$$

where $\alpha = p(s = 1)$.

- In cases (i) and (ii), GIW is reduced to IW since $\alpha = 1$.

Theorem 4. *GIW is always risk-consistent for distribution shift problems.*

Implementation of GIW

- Split validation data and estimate α
  - Pretrain on training data to obtain a feature extractor
  - Train a one-class SVM[3] on latent representation of training data
  - Split validation data into IT $\{(\boldsymbol{x}_i^{\mathrm{v}1}, y_i^{\mathrm{v}1})\}_{i=1}^{n_{\mathrm{v}1}}$ & OOT $\{(\boldsymbol{x}_i^{\mathrm{v}2}, y_i^{\mathrm{v}2})\}_{i=1}^{n_{\mathrm{v}2}}$ parts by the one-class SVM
  - Estimate α as $\hat{\alpha} = \frac{n_{\mathrm{v}1}}{n_{\mathrm{v}}}$
- Empirical objective of GIW is:

$$\hat{J}_G(\boldsymbol{f}) = \frac{n_{\mathrm{v}1}}{n_{\mathrm{v}} n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}\left(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}}\right) \ell\left(\boldsymbol{f}\left(\boldsymbol{x}_i^{\mathrm{tr}}\right), y_i^{\mathrm{tr}}\right) + \frac{1}{n_{\mathrm{v}}} \sum_{j=1}^{n_{\mathrm{v}2}} \ell\left(\boldsymbol{f}\left(\boldsymbol{x}_j^{\mathrm{v}2}\right), y_j^{\mathrm{v}2}\right).$$

[3] B. Schölkopf et al., Support vector method for novelty detection. In NeurIPS, 1999.

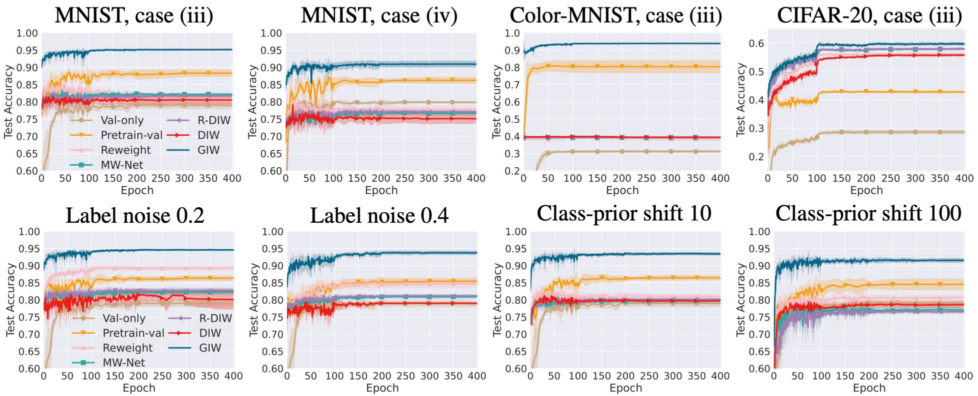
Experiments

- Two distribution shift (DS) patterns
  - Support shift*:* DS solely comes from the support mismatch
  - Support-distribution shift: add DS on top of the support shift
- Setups & results*

Dataset	Task (classification for)	Training data	Test data	Model
MNIST	odd and even digits	4 digits (0-3)	10 digits (0-9)*	LeNet-5
Color-MNIST	10 digits	digits all in red	digits in red/blue/green	LeNet-5
CIFAR-20	objects in 20 superclasses	2 classes/superclass	5 classes/superclass	ResNet-18

Support shift

Support-distribution shift
(i.e., label noise/
class-prior shift)



* Results of support-distribution shift here are on MNIST; results on more datasets/baselines can be found in the paper.

Thanks for your attention!