uai2024

Differentiable Pareto-Smoothed Weighting for High-Dimensional Heterogeneous Treatment Effect Estimation

Yoichi Chikahara[1], Kansei Ushiyama[2]. [1]NTT. [2]The University of Tokyo

Problem setup: CATE estimation from high-dimensional observational data

Input Observational data $D \overset{\text{i.i.d.}}{\sim} \mathrm{P}(A, \boldsymbol{X}, Y)$

Treatment $A \in \{0,1\}$ | Features $\boldsymbol{X}$ | Outcome Y | Potential outcomes Y^a

A	X_1	…	X_d	Y	Y^0	Y^1
0		…		0.2	0.2	?
1		…		1.1	?	1.1

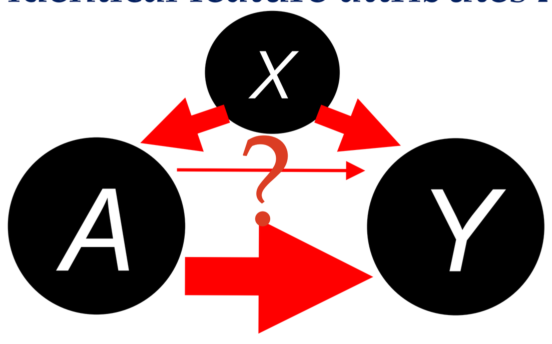
High-dimensional

$Y = AY^1 + (1-A)Y^0$ Observed when $A = a$

Output *Conditional Average Treatment Effect (CATE)*

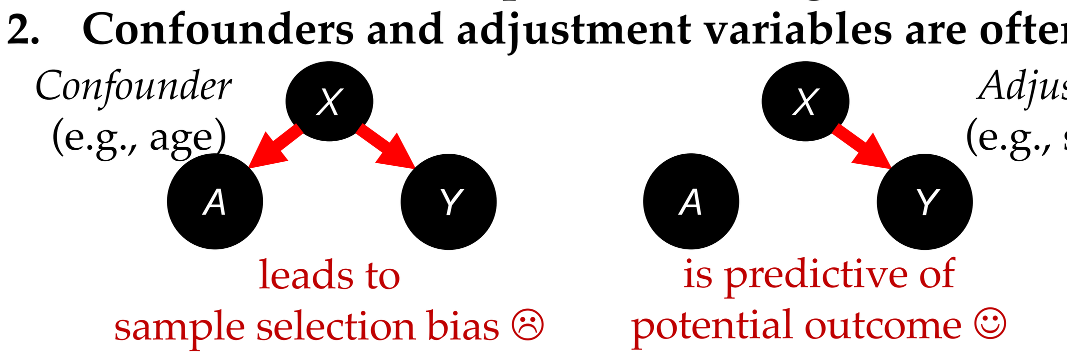
$$\mathrm{CATE}(\boldsymbol{x}) \coloneqq \mathbb{E}[Y^1 - Y^0 \mid \boldsymbol{X} = \boldsymbol{x}]$$

Average treatment effect across individuals with identical feature attributes $\boldsymbol{X} = \boldsymbol{x}$



Difficulty in high-dimensional setup:

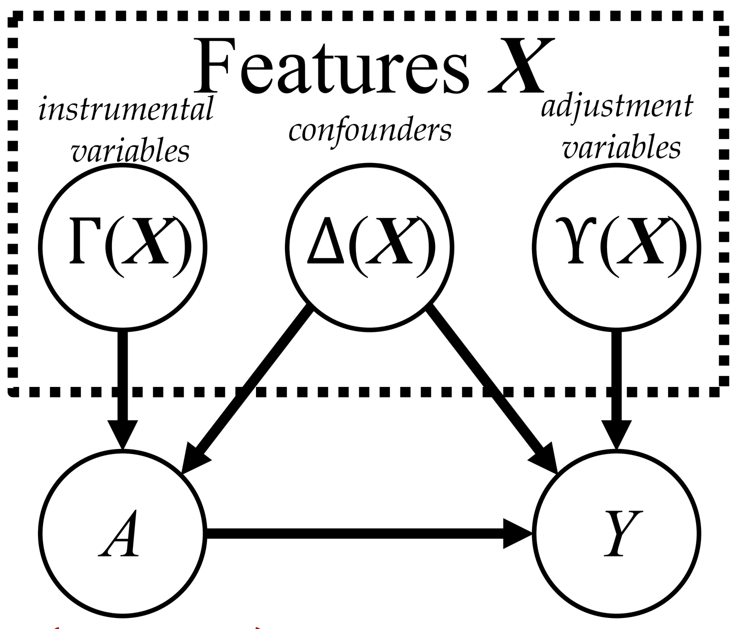
1. **We often have little prior knowledge about features $\boldsymbol{X}$**
2. **Confounders and adjustment variables are often *intertwined***



Question:
Can we remove sample selection bias due to *confounders* while keeping predictive information of *adjustment variables*?

Weighted representation learning

Existing Approach: **Data-driven feature separation**



$A \sim \pi(\Gamma(\boldsymbol{x}), \Delta(\boldsymbol{x}))$ $\hat{Y}^a = h^a(\Delta(\boldsymbol{x}), \Upsilon(\boldsymbol{x}))$

1. Fit propensity score $\pi(\cdot)$ by

$$\min_{\pi} \underbrace{-\frac{1}{n}\sum_{i=1}^{n}\Big(a_i \log(\pi(\Gamma(\boldsymbol{x}_i), \Delta(\boldsymbol{x}_i))) + (1-a_i)\log(1-\pi(\Gamma(\boldsymbol{x}_i), \Delta(\boldsymbol{x}_i)))\Big)}_{\text{Binary cross entropy loss}} + \lambda_\pi \Omega(\pi)$$

2. Train $\Gamma(\cdot), \Delta(\cdot), \Upsilon(\cdot), h^0(\cdot), h^1(\cdot)$ by

$$\min_{\Gamma,\Delta,\Upsilon,h^0,h^1} \underbrace{\frac{1}{n}\sum_{i=1}^{n} w_i l(y_i, h^{a_i}(\Delta(\boldsymbol{x}_i), \Upsilon(\boldsymbol{x}_i)))}_{\text{Weighted prediction loss}} + \underbrace{\lambda_\Upsilon \mathrm{MMD}\left(\{\Upsilon(\boldsymbol{x}_i)\}_{i:a_i=0}, \{\Upsilon(\boldsymbol{x}_i)\}_{i:a_i=1}\right)}_{\text{Penalize dependence of } \Upsilon(\boldsymbol{X}) \text{ on } A} + \lambda_{-\pi}\Omega\left(\Gamma, \Delta, \Upsilon, h^0, h^1\right)$$

where

$$w_i = \frac{\mathrm{P}(\Gamma(\boldsymbol{x}_i), \Delta(\boldsymbol{x}_i) \mid A = a_i)}{\mathrm{P}(\Gamma(\boldsymbol{x}_i), \Delta(\boldsymbol{x}_i) \mid A = a_i)} + \frac{\mathrm{P}(\Gamma(\boldsymbol{x}_i), \Delta(\boldsymbol{x}_i) \mid A = 1 - a_i)}{\mathrm{P}(\Gamma(\boldsymbol{x}_i), \Delta(\boldsymbol{x}_i) \mid A = a_i)} \propto \frac{1}{\mathrm{P}(A = a_i \mid \Gamma(\boldsymbol{x}_i), \Delta(\boldsymbol{x}_i))} \coloneqq \frac{1}{\pi_{a_i}(\Gamma(\boldsymbol{x}_i), \Delta(\boldsymbol{x}_i))}$$

Weakness: Inverse probability weight w_i is numerically unstable:
Even slight propensity score estimation error leads to large CATE estimation error

Weight smoothing with Pareto smoothing

Advantage:
1. **Can obtain a less biased estimator than weight truncation**
2. **Can be combined with self-normalization**

Main idea Improve CATE estimation stability by Pareto smoothing

- Pareto smoothing [Vehtari+; JMLR2024]: Replace the $M+1$ largest importance sampling weight values with inverse CDF of generalized Pareto distribution

$$w_{[i]} = \mathbf{I}(i \geq n - M + 1)\,\hat{\mathrm{F}}^{-1}\left(\frac{i - (n-M) - 1/2}{M}\right) + (1 - \mathbf{I}(i \geq n - M + 1))\, w_{[i]}$$

where $w_{[1]} \leq \cdots \leq w_{[n]}$, $M = \min\left\{\left\lfloor \frac{n}{5} \right\rfloor, \lfloor 3\sqrt{n} \rfloor\right\}$, and $\mathrm{F}(w) = \begin{cases} 1 - \left(1 + \frac{\xi(w-\mu)}{\sigma}\right)^{-\frac{1}{\xi}} & (\xi \neq 0), \\ 1 - \mathrm{e}^{-\frac{w-\mu}{\sigma}} & (\xi = 0) \end{cases}$

- Need to compute rank $\boldsymbol{r} = r(\boldsymbol{w})$:
  - Example: If $w_3 \leq w_1 \leq w_2$, since $w_1 = w_{[2]}$, $w_2 = w_{[3]}$, $w_3 = w_{[1]}$, $\boldsymbol{r} = [2, 3, 1]$

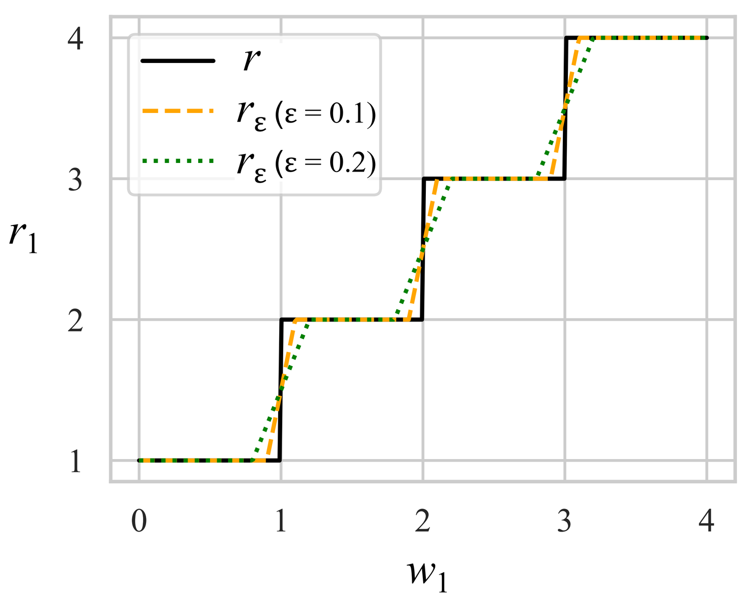
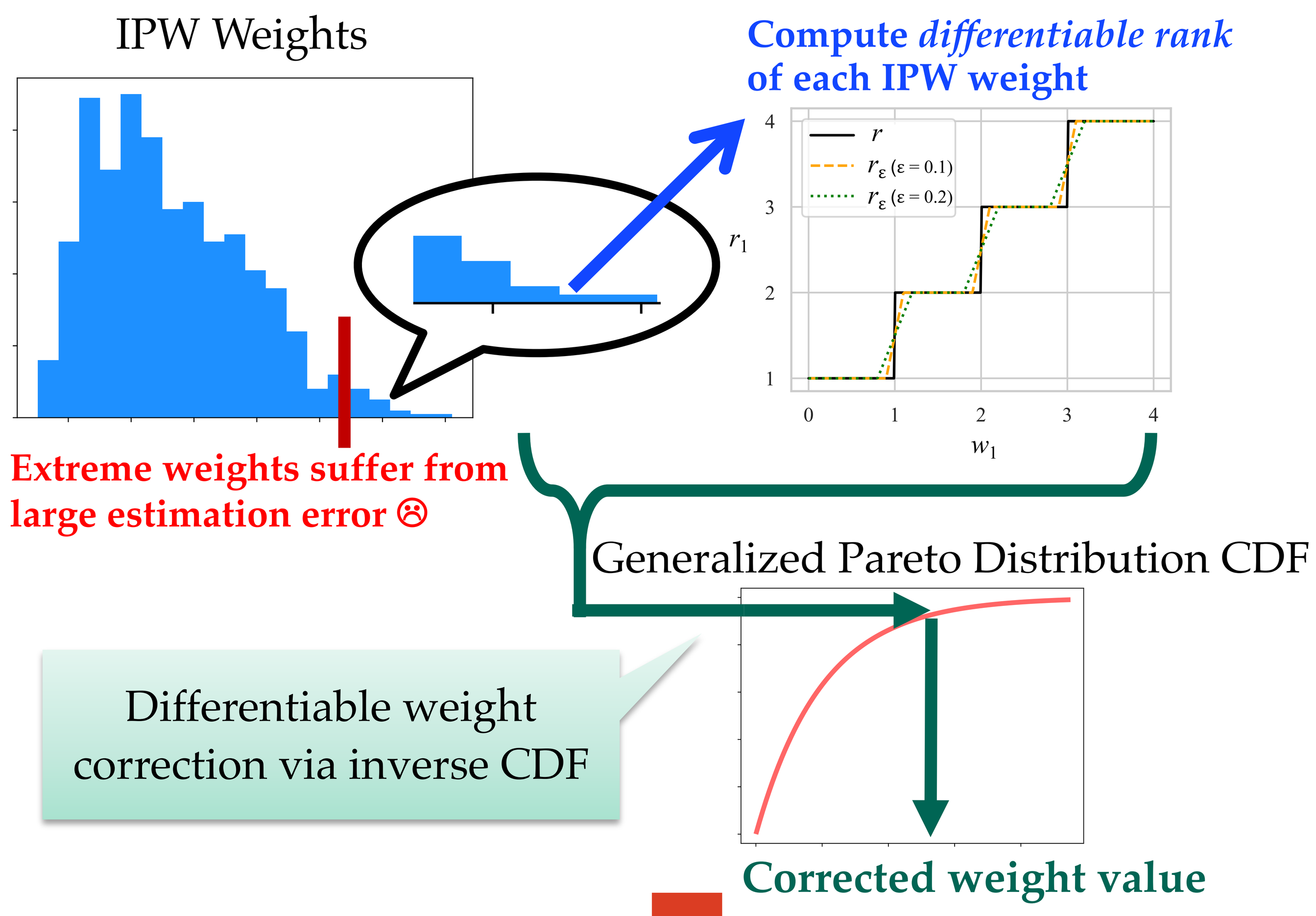


Figure 2: Illustration of rank function $\boldsymbol{r} = r(\boldsymbol{w})$ (black) and differentiable rank functions $\boldsymbol{r} = r_\varepsilon(\boldsymbol{w})$ (orange and green). Here we take input vector $\boldsymbol{w} = [w_1, 1, 2, 3]^\top$, vary w_1's value, and look at how its rank $r_1 \in \boldsymbol{r}$ changes. When regularization parameter $\varepsilon \to 0$, r_ε converges to r [Blondel et al., 2020].

Difficulty: $r(\boldsymbol{w})$ is piecewise constant: Gradient is always zero or undefined.
We cannot perform gradient back propagation ☹

Proposed method

Proposed method: **Pareto smoothing + Differentiable ranking**



End-to-end weighted representation learning for CATE estimation from high-dimensional observational data

1. Approximate non-differentiable rank function r with differentiable one
   - Fast soft rank [Blondel+; ICML2020]: Approximate as a solution to regularized LP
2. Approximate indicator function with sigmoid

$$\mathbf{I}(i \geq j) \simeq \varsigma(i, j) \coloneqq \frac{1}{1 + \mathrm{e}^{-\kappa(i-j)}}$$

Combining 1. & 2. leads to the following weight replacement formula:

$$\tilde{w}_i = \varsigma(r_i, n - M + 1)\,\tilde{\mathrm{F}}^{-1}\left(\zeta\left(\frac{r_i - (n - M) - 1/2}{M}\right)\right) + (1 - \varsigma(r_i, n - M + 1))\, w_i$$

where $\zeta(x) \coloneqq \min\{\max\{x, 0\}, 1\}$

Algorithm 1 Differentiable Pareto-Smoothed Weighting (DPSW)

Initialize the parameters of Γ, Δ, Υ, π, h^0, and h^1
while not converged **do**
  while not converged **do**
    Sample mini-batch from $\mathcal{D} = \{(a_i, \boldsymbol{x}_i, y_i)\}_{i=1}^n$
    Update π by minimizing cross entropy loss in (2)
  end while
  while not converged **do**
    Sample mini-batch from $\mathcal{D} = \{(a_i, \boldsymbol{x}_i, y_i)\}_{i=1}^n$
    for instance i in mini-batch **do**
      Compute weight w_i by (4)
    end for
    Compute differentiable rank $\boldsymbol{r} = r_\varepsilon(\boldsymbol{w})$
    Estimate GPD parameters as $\tilde{\mu}$, $\tilde{\sigma}$, and $\tilde{\xi}$
    for instance i in mini-batch **do**
      Replace each weight w_i with $\tilde{w}_i$ in (20)
    end for
    Update Γ, Δ, Υ, h^0, and h^1 by minimizing prediction loss in (3) with Pareto-smoothed weights $\{\tilde{w}_i\}$
  end while
end while

Experimental results:

Semi-synthetic data

Table 1: Mean and standard deviation of test PEHE on semi-synthetic datasets (Lower is better)

Method	News ($d = 3477$)	ACIC ($d = 177$)
LR-1	3.35 ± 1.28	0.72 ± 0.07
LR-2	5.36 ± 1.75	3.82 ± 0.15
SL	2.83 ± 1.11	1.69 ± 0.52
TL	2.55 ± 0.82	2.23 ± 0.50
XL	2.77 ± 1.01	1.05 ± 0.72
DRL	23.9 ± 5.96	3.77 ± 8.96
CF	3.84 ± 1.67	3.55 ± 0.19
CF DML	2.69 ± 1.06	1.18 ± 0.32
TARNet	4.92 ± 1.80	3.31 ± 0.51
GANITE	2.68 ± 0.66	3.69 ± 0.77
DRCFR	2.38 ± 0.66	0.98 ± 0.07
DRCFR Norm.	2.37 ± 0.94	0.73 ± 0.12
DRCFR Trunc.	2.42 ± 0.79	1.06 ± 0.06
PSW	4.03 ± 1.35	0.71 ± 0.01
DPSW	**2.20 ± 0.72**	**0.57 ± 0.03**
DPSW Norm.	**2.10 ± 0.66**	**0.52 ± 0.16**

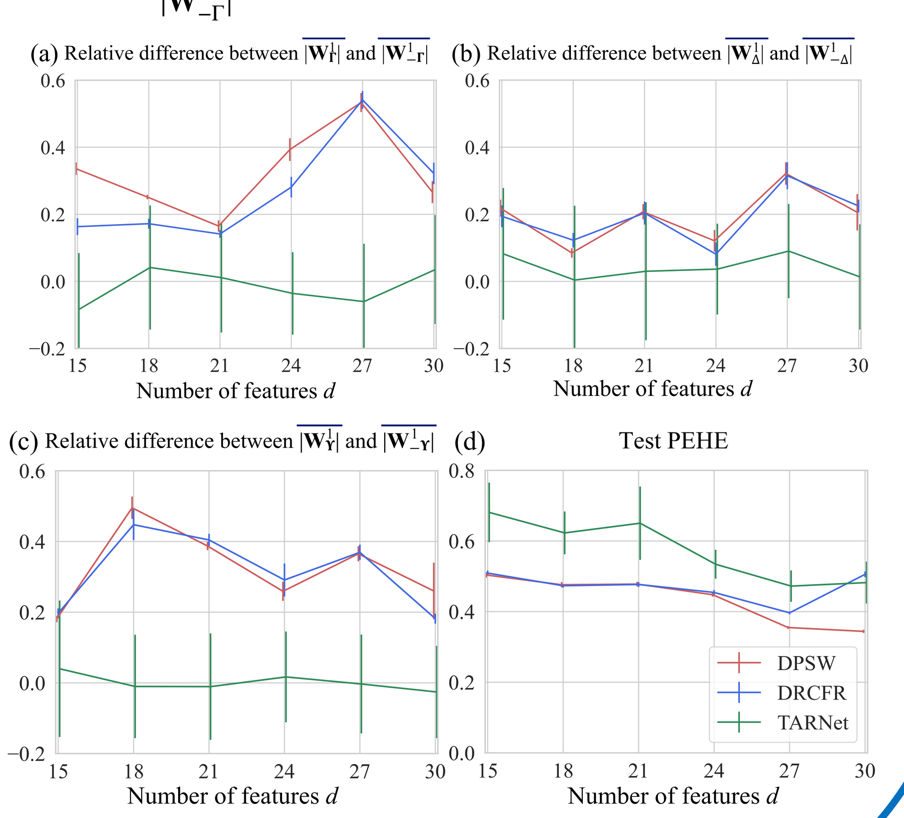
Synthetic data

Randomly generate features as

$[X_\Gamma, X_\Delta, X_\Upsilon]^\top \in \mathbb{R}^d$ ($d = 15, 18, \ldots, 30$)

Measure the relative difference of average absolute values of the first-layered weight submatrices, e.g.,

$$\frac{\overline{|\mathbf{W}^1_\Gamma|} - \overline{|\mathbf{W}^1_{-\Gamma}|}}{\overline{|\mathbf{W}^1_{-\Gamma}|}} \quad \text{where} \quad \mathbf{W}^1 = [\mathbf{W}^1_\Gamma, \mathbf{W}^1_{-\Gamma}]$$



References arXiv link: GitHub code link: Web page link:

E-mail: chikahara.yoichi @ gmail.com