RieszNet and ForestRiesz: Automatic Debiased Machine Learning with Neural Nets and Random Forests

Víctor Quintas-Martínez (MIT)

ICML 2022

Joint with Victor Chernozhukov (MIT), Whitney Newey (MIT) and Vasilis Syrgkanis (MS Research)

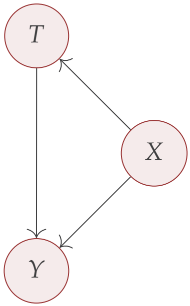
Examples in Causal Inference

Massachusetts Institute of Technology

EXAMPLE 1: Average Treatment Effect of binary treatment

- Suppose that we want to estimate the causal impact of a treatment $T \in \{0, 1\}$ on an outcome Y
- In observational settings, this type of inference is complicated by the presence of confounders that affect both T and Y
- However, if we have access to a rich enough set of covariates X such that the treatment is as good as randomly assigned conditional on those covariates, we might still be able to identify an ATE:

$$\theta_0 := \mathrm{E}\left[\mathrm{E}\left[Y \mid T = 1, X\right] - \mathrm{E}\left[Y \mid T = 0, X\right]\right]$$

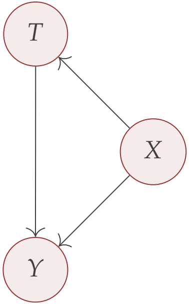


Examples in Causal Inference

Example 2: Average Derivative of continuous treatment

- When T is continuous, we may be interested in estimating an average derivative or average marginal effect

$$\theta_0 := \mathrm{E}\left[\partial_T \mathrm{E}\left[Y \mid T, X\right]\right]$$



General Setting

Massachusetts Institute of Technology

- We want to provide a point estimate and a confidence interval for:

$$\theta_0 := \mathrm{E}\left[m(W;\gamma_0)\right]$$

where $W := (Y, Z)$, $Z := (T, X)$ and $\gamma_0(Z) := \mathrm{E}\left[Y \mid Z\right]$ is an (unknown) regression function

General Setting

- We want to provide a point estimate and a confidence interval for:

$$\theta_0 := \mathrm{E}\left[m(W;\gamma_0)\right]$$

where $W := (Y,Z)$, $Z := (T,X)$ and $\gamma_0(Z) := \mathrm{E}\left[Y \mid Z\right]$ is an (unknown) regression function

- We want to use a ML estimator $\widehat{\gamma}$, but because of regularization and/or model selection, the direct estimator:

$$\hat{\theta}_{\mathrm{direct}} := \mathbb{E}_n\left[m(W;\widehat{\gamma})\right]$$

may have a bias that vanishes at a $\sqrt{n}$ rate or slower, and may not even be asymptotically normal

- This invalidates usual CIs based on asymptotic normality

Debiased Machine Learning

Massachusetts Institute of Technology

- We want to construct a *debiased* ML estimator:

$$\hat{\theta}_{\mathrm{DML}} := \mathbb{E}[m(W;\widehat{\gamma}) + \underbrace{\hat{\alpha}(Z)(Y - \widehat{\gamma}(Z))}_{\text{debiasing term}}]$$

- But what should this $\hat{\alpha}$ be?

Debiased Machine Learning

Massachusetts Institute of Technology

- We want to construct a *debiased* ML estimator:

$$\hat{\theta}_{\mathrm{DML}} := \mathbb{E}[m(W;\widehat{\gamma}) + \underbrace{\hat{\alpha}(Z)(Y - \widehat{\gamma}(Z))}_{\text{debiasing term}}]$$

- But what should this $\hat{\alpha}$ be?

Lemma (Riesz Representation Theorem)

If $\gamma \mapsto \mathrm{E}\left[m(W;\gamma)\right]$ *is a continuous linear functional, then there exists* α_0 *(Riesz representer, RR) such that*

$$\mathrm{E}\left[m(W;\gamma)\right] = \mathrm{E}\left[\alpha_0(Z)\gamma(Z)\right]$$

for all γ *with* $\mathrm{E}\left[\gamma(Z)^2\right] < \infty$.

Debiased Machine Learning

- We want to construct a *debiased* ML estimator:

$$\hat{\theta}_{\mathrm{DML}} := \mathbb{E}[m(W;\widehat{\gamma}) + \underbrace{\hat{\alpha}(Z)(Y - \widehat{\gamma}(Z))}_{\text{debiasing term}}]$$

- But what should this $\hat{\alpha}$ be?

- The RR exists in Examples 1 and 2 under mild regularity conditions:

  EXAMPLE 1: α_0 is the Horvitz-Thompson transformation:

$$\alpha_0(T, X) = T/\Pr(T = 1 \mid X) - (1 - T)/(1 - \Pr(T = 1 \mid X))$$

  EXAMPLE 2: α_0 is a generalized propensity score:

$$\alpha_0(T, X) = -\partial_t \log f(T \mid X)$$

Debiased Machine Learning

Massachusetts Institute of Technology

- We want to construct a *debiased* ML estimator:

$$\hat{\theta}_{\text{DML}} := \mathbb{E}[m(W;\widehat{\gamma}) + \underbrace{\hat{\alpha}(Z)(Y - \widehat{\gamma}(Z))}_{\text{debiasing term}}]$$

- But what should this $\hat{\alpha}$ be?

- The augmented moment satisfies a *mixed bias* property:

$$\mathrm{E}\left[m(W;\gamma) + \alpha(Z)(Y - \gamma(Z))\right] = \theta_0 - \mathrm{E}\left[(\alpha(Z) - \alpha_0(Z))(\gamma(Z) - \gamma_0(Z))\right]$$

- If $\sqrt{n}\|\hat{\alpha} - \alpha_0\|_{L^2}\|\widehat{\gamma} - \gamma_0\|_{L^2} \to 0$, then asymptotic normality is restored

$$\sqrt{n}(\hat{\theta}_{\text{DML}} - \theta_0) \Rightarrow N(0, V)$$

where $V = \mathrm{Var}\left\{m(W;\gamma_0) + \alpha_0(Z)(Y - \gamma_0(Z))\right\}$

Riesz Representer as the Minimizer of a Stochastic Loss

- The first generation of debiased ML estimators used the explicit form of the RR

  EXAMPLE 1: Estimate the propensity score $\Pr(T = 1 \mid X)$ and plug it in the RR formula (AIPW estimator)

Riesz Representer as the Minimizer of a Stochastic Loss

- Here, instead, we use the fact that:

$$\begin{aligned}\alpha_0 &= \arg\min_{\alpha} \mathrm{E}\left[\alpha(Z)^2 - 2m(W;\alpha)\right] \\ &= \arg\min_{\alpha} \mathrm{E}\left[\alpha(Z)^2 - 2\alpha_0(Z)\alpha(Z) + \alpha_0(Z)^2\right] \\ &= \arg\min_{\alpha} \mathrm{E}\left[\left(\alpha(Z) - \alpha_0(Z)\right)^2\right]\end{aligned}$$

  to estimate the RR by the empirical analogue:

$$\hat{\alpha} = \arg\min_{\alpha\in\mathcal{A}_n} \mathbb{E}_n\left[\alpha(Z)^2 - 2m(W;\alpha)\right] \qquad (*)$$

- Automatic approach in that it relies only on black-box evaluation oracle access to the linear functional and does not require knowledge of the analytic form of α_0

Lemma

To estimate $\mathrm{E}\left[m(W;\gamma_0)\right]$ *it suffices to consider regression functions that condition only on the value of the RR, i.e.* $\gamma_0(Z) = h_0(\alpha_0(Z))$

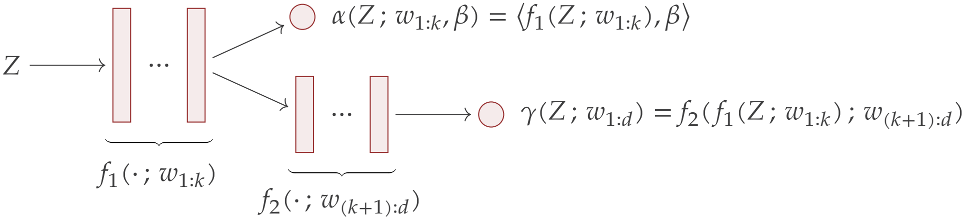
RieszNet: Targeted Regularization and Multitasking

Architecture

Lemma

To estimate $\mathrm{E}[m(W;\gamma_0)]$ *it suffices to consider regression functions that condition only on the value of the RR, i.e.* $\gamma_0(Z) = h_0(\alpha_0(Z))$

- Based on this Lemma, we consider a deep neural representation of the RR and the regression as follows:



RieszNet: Targeted Regularization and Multitasking

Targeted Regularization

- Inspired by the TMLE framework (Bang & Robins, 2005; Van der Laan et al., 2021), we consider a corrected regression:

  $$\tilde{\gamma}(Z) = \gamma(Z) + \epsilon \cdot \alpha(Z),$$

  where ϵ is the OLS coefficient of $Y - \gamma(Z)$ on $\alpha(Z)$

- The parameter ϵ is optimized together with the rest of the network (as in dragonnet, Shi et al., 2019), rather than in a post-processing step

Massachusetts Institute of Technology

- Our multitasking architecture minimizes the combined loss:

$$\min_{w_{1:d},\beta,\epsilon} \mathrm{REGloss}(w_{1:d}) + \lambda_1 \mathrm{RRloss}(w_{1:k},\beta) + \lambda_2 \mathrm{TMLEloss}(w_{1:d},\beta,\epsilon) + R(w_{1:d},\beta)$$

  where:

$$\begin{aligned} \mathrm{REGloss}(w_{1:d}) &:= \mathbb{E}_n\left[(Y - \gamma(Z; w_{1:d}))^2\right] \\ \mathrm{RRloss}(w_{1:k},\beta) &:= \mathbb{E}_n\left[\alpha(Z; w_{1:k},\beta)^2 - 2\, m(W; \alpha(\cdot; w_{1:k},\beta))\right] \\ \mathrm{TMLEloss}(w_{1:d},\beta,\epsilon) &:= \mathbb{E}_n\left[(Y - \gamma(Z; w_{1:d}) - \epsilon \cdot \alpha(Z; w_{1:k},\beta))^2\right] \end{aligned}$$

  and $R(w_{1:d},\beta)$ is a penalty that does not take ϵ as input

- We train the weights by minimizing this loss with stochastic first-order methods

ForestRiesz: Locally Linear Riesz Estimation

Sieve Parametrization

- One approach to estimating α_0 by regression trees would be to allow splits with respect to all input variables $Z = (T, X)$
  - However, this approach could introduce large discontinuities in T, under which our asymptotic theory is not valid

ForestRiesz: Locally Linear Riesz Estimation

Sieve Parametrization

- One approach to estimating α_0 by regression trees would be to allow splits with respect to all input variables $Z = (T, X)$
  - However, this approach could introduce large discontinuities in T, under which our asymptotic theory is not valid

- Instead, we parametrize $\alpha(Z)$ as a locally linear function:

$$\alpha(Z) = \langle \phi_\alpha(T, X), \beta_\alpha(X) \rangle ,$$

where $\phi_\alpha(T, X)$ is a (smooth) pre-defined feature map and $\beta_\alpha(X)$ is a non-parametric component estimated based on the tree splits

ForestRiesz: Locally Linear Riesz Estimation

Estimation by GRF

- The non-parametric component β_α minimizes the Riesz loss:

$$\min_{\beta_\alpha} \mathrm{E}\left[\beta_\alpha(x)^\top \phi_\alpha(Z)\phi_\alpha(Z)^\top \beta_\alpha(x) - 2\,\beta_\alpha(x)^\top m(W;\phi_\alpha) \mid X = x\right]$$

  which admits the following local first order condition:

$$\mathrm{E}\left[\phi_\alpha(Z)\phi_\alpha(Z)^\top \beta_\alpha(x) - m(W;\phi_\alpha) \mid X = x\right] = 0$$

ForestRiesz: Locally Linear Riesz Estimation

Estimation by GRF

- The non-parametric component β_α minimizes the Riesz loss:

$$\min_{\beta_\alpha} \mathrm{E}\left[\beta_\alpha(x)^\top \phi_\alpha(Z)\phi_\alpha(Z)^\top \beta_\alpha(x) - 2\,\beta_\alpha(x)^\top m(W;\phi_\alpha) \mid X = x\right]$$

  which admits the following local first order condition:

$$\mathrm{E}\left[\phi_\alpha(Z)\phi_\alpha(Z)^\top \beta_\alpha(x) - m(W;\phi_\alpha) \mid X = x\right] = 0$$

- This falls into the class of problems defined by solutions to moment conditions considered in the Generalized Random Forests framework of Athey et al. (2019)
  - We modify the original GRF heterogeneity criterion to maximize a version weighted by the local Jacobians $J(\mathrm{child}) = |\mathrm{child}|^{-1} \sum_{i\in\mathrm{child}} \phi_\alpha(Z_i)\phi_\alpha(Z_i)^\top$
  - We want to penalize splits where the covariance matrix of the feature map is ill-conditioned

ForestRiesz: Locally Linear Riesz Estimation

Regression

- We can do the same for the regression function:

$$\min_{\beta_\gamma} \mathrm{E}\left[(Y - \phi_\gamma(Z)^\top \beta_\gamma(X))^2\right]$$

  with local first order condition:

$$\mathrm{E}\left[\phi_\gamma(Z)\phi_\gamma(Z)^\top \beta_\gamma(x) - \phi_\gamma(Z)Y \mid X = x\right] = 0$$

- In fact, we can even build a multitasking version of ForestRiesz where we stack the moment conditions

Results: Average Treatment Effect in the IHDP Dataset

- IHDP was an experiment designed to evaluate the effect of home visits and attendance at specialized clinics T on future developmental outcomes Y of low birth weight infants
- $n = 747$, $\dim(X) = 25$ continuous and binary covariates
- Taking X from the data, generate T and 1000 synthetic draws of Y with the `NPCI R` package, same setting as Shi et al. (2019) for comparability

Results: Average Treatment Effect in the IHDP Dataset

Table: Mean Absolute Error (MAE) and its standard error over 1000 semi-synthetic datasets based on the IHDP experiment

(a) RieszNet

	MAE ± std. err.
RieszNet	0.110 ± 0.003
Benchmark: Dragonnet (Shi et al., 2019)	0.146 ± 0.010

(b) ForestRiesz

	MAE ± std. err.
ForestRiesz	0.126 ± 0.004
Benchmark: CausalForest (Athey et al., 2019)	0.728 ± 0.028

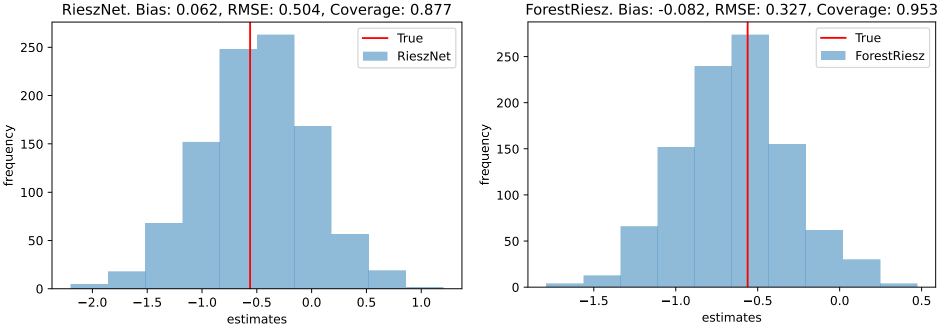
Results: Average Derivative in the BHP Gasoline Demand Data

- Gasoline demand data from 2001 National Household Travel Survey (Blundell et al., 2017). Want to estimate average derivative of $Y = \log(\text{quantity})$ with respect to $T = \log(\text{price})$
- $n = 3466$, $\dim(X) = 50$ continuous and binary covariates, including household characteristics and geographic controls
- Take X and estimate $\mu_T(X) := \mathrm{E}[T \mid X]$, $\sigma_T^2(X) := \mathrm{Var}(T \mid X)$ from the data
- Draw $T \sim N(\mu_T(X), \sigma_T^2(X))$ and generate $Y = f(T, X) + \varepsilon$. Here we show the most complex $f(\cdot)$ with linear and non-linear confounding

Results: Average Derivative in the BHP Gasoline Demand Data

Figure: RieszNet and ForestRiesz: bias, RMSE, coverage and distribution of estimates over 1000 semi-synthetic datasets based on the BHP gasoline demand data

(a) RieszNet (b) ForestRiesz + post-TMLE



Ablation Studies: RieszNet

Effect of Multitasking and End-to-End Learning

- Row 2 uses no multitasking, the Riesz representer and regression function are estimated using separate NNs

Table: IHDP ablation studies for RieszNet

	RieszNet		
	Bias	RMSE	Cov.
Baseline	-0.044	0.147	0.950
Separate NNs	-0.176	0.411	0.880
No end-to-end	-0.051	1.221	0.650
TMLE post-proc.	-0.088	0.182	0.950

Ablation Studies: RieszNet

Effect of Multitasking and End-to-End Learning

- Row 3 removes "end-to-end" training of the shared representation: the weights of the common layers are trained on the Riesz loss only, then frozen when optimizing the regression loss

Table: IHDP ablation studies for RieszNet

	RieszNet		
	Bias	RMSE	Cov.
Baseline	-0.044	0.147	0.950
Separate NNs	-0.176	0.411	0.880
No end-to-end	-0.051	1.221	0.650
TMLE post-proc.	-0.088	0.182	0.950

Ablation Studies: RieszNet

Effect of Multitasking and End-to-End Learning

- Row 4 removes "end-to-end" learning of the TMLE adjustment: we set $\lambda_2 = 0$ and then adjust the outputs of RieszNet in a standard TMLE post-processing step

Table: IHDP ablation studies for RieszNet

	RieszNet		
	Bias	RMSE	Cov.
Baseline	-0.044	0.147	0.950
Separate NNs	-0.176	0.411	0.880
No end-to-end	-0.051	1.221	0.650
TMLE post-proc.	-0.088	0.182	0.950

Ablation Studies: ForestRiesz

Effect of Multitasking and Cross-fitting

- Cross-fitting: split the sample in folds $\ell = 1, \ldots 5$. For each ℓ, use the data *not* in ℓ to obtain $\widehat{\gamma}_{-\ell}$ and $\hat{\alpha}_{-\ell}$, and then use the data *in* ℓ to estimate the average moment
- Double cross-fitting: the same, but $\widehat{\gamma}_{-\ell}$ and $\hat{\alpha}_{-\ell}$ are estimated using different sub-samples

Table: BHP ablation studies for ForestRiesz

	ForestRiesz + post-TMLE		
	Bias	RMSE	Cov.
Baseline (x-fit, m-task)	-0.082	0.327	0.953
No x-fit, no m-task	-0.079	0.314	0.827
No x-fit, m-task	-0.060	0.326	0.835
X-fit, no m-task	-0.091	0.331	0.945
Double x-fit	-0.094	0.338	0.950

Summing Up

- Provide the first Auto-DML implementation using Neural Nets (RieszNet) and Random Forests (ForestRiesz)
  - Theory guarantees for generic Auto-DML in Chernozhukov et al. (2021)
- Experimentally evaluate the proposed methods in two settings (ATE and average derivative)
  - Find superior performance to benchmarks
  - Ablation studies to demonstrate which features of our estimators are crucial for the gains

Thank you for watching!