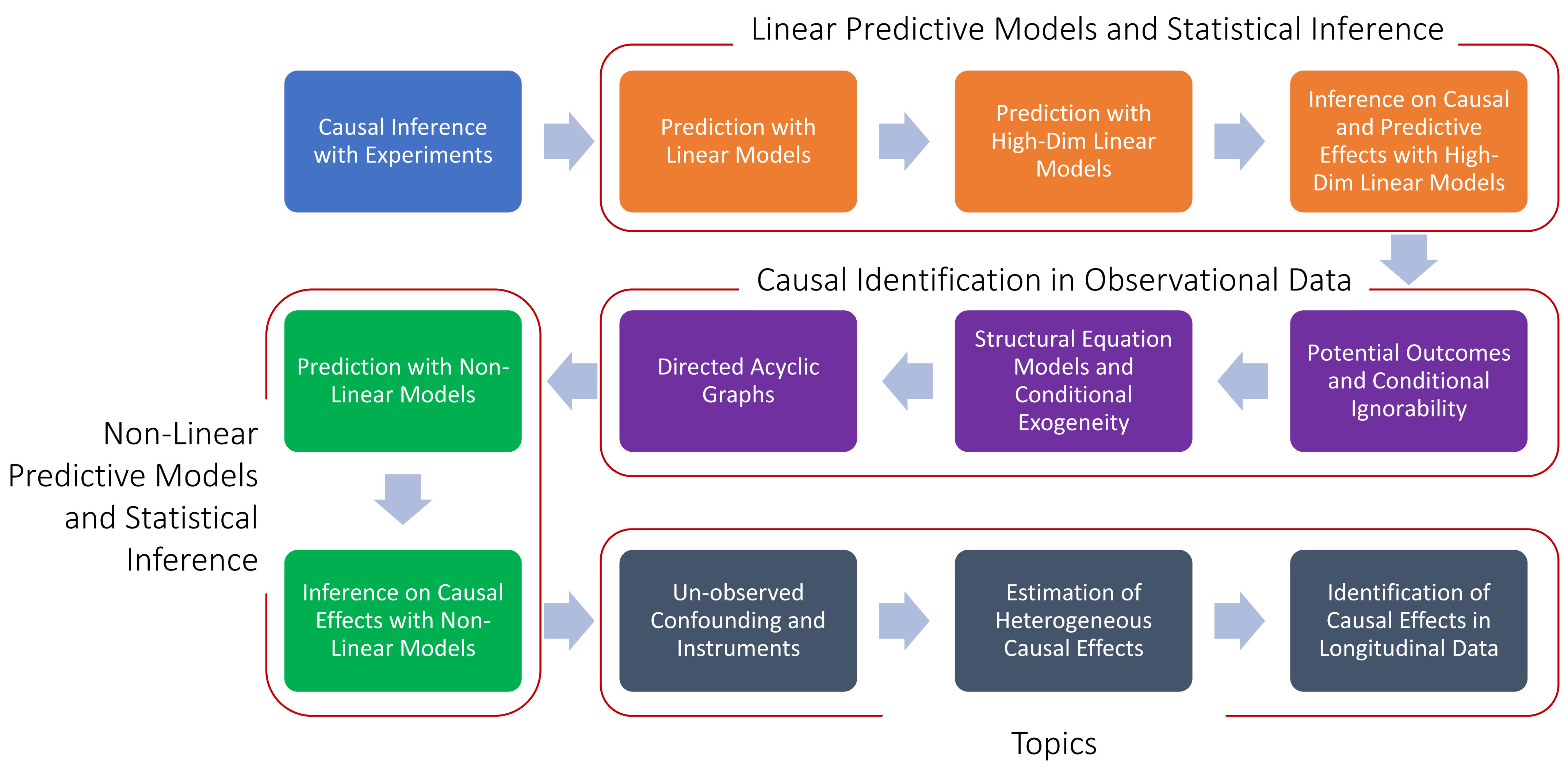
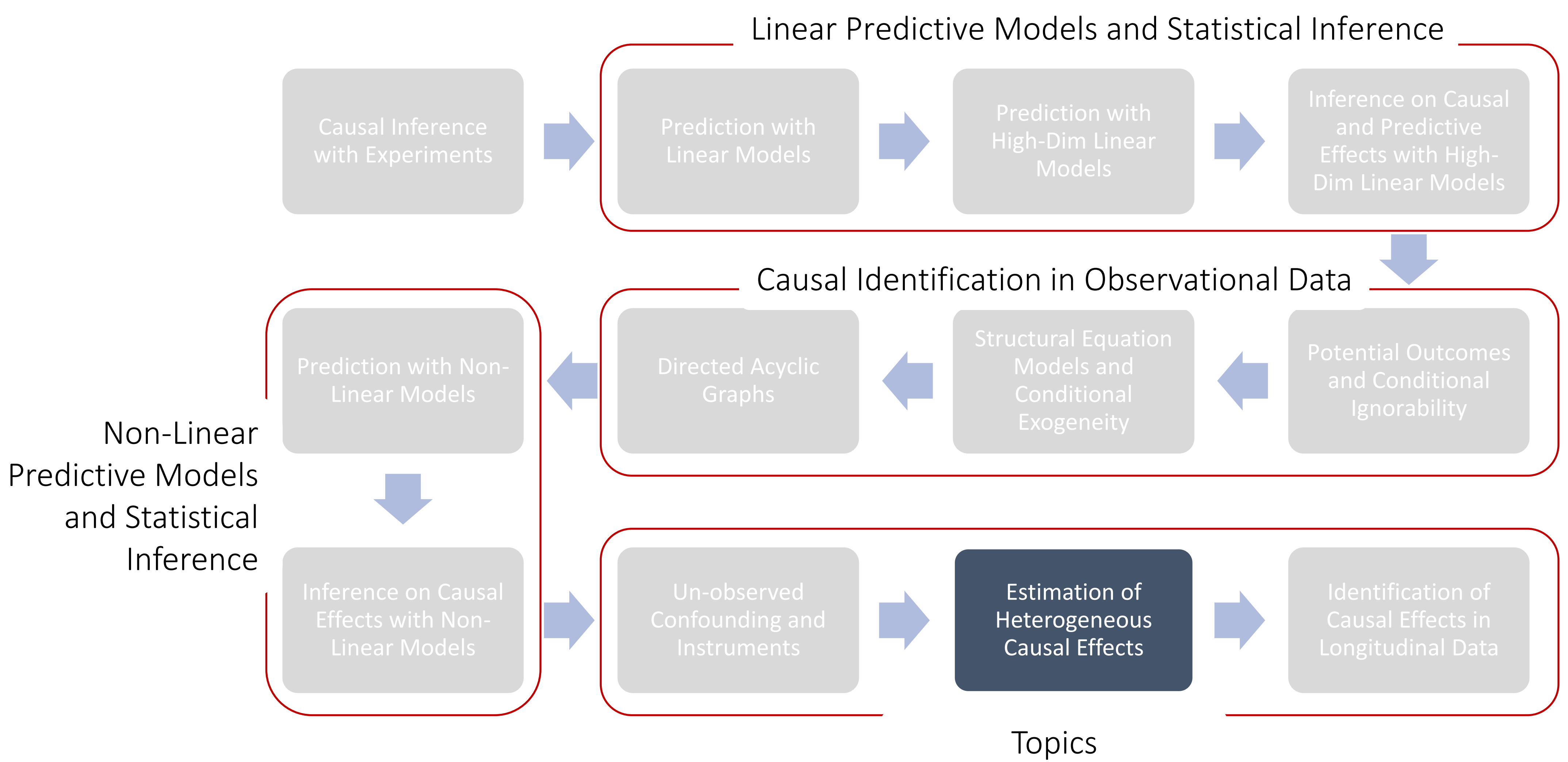
MS&E 228: Unobserved Confounding and Instruments

Vasilis Syrgkanis

MS&E, Stanford

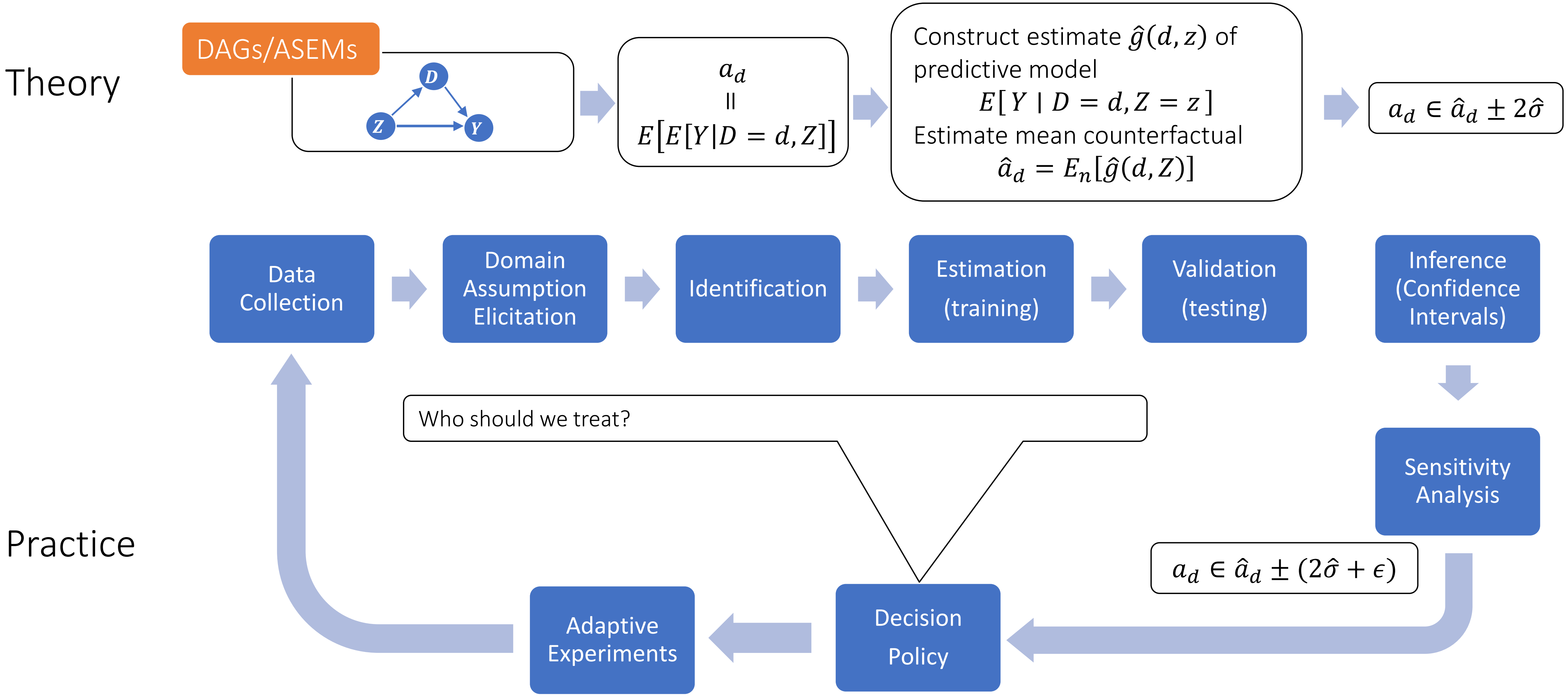

Linear Predictive Models and Statistical Inference
Causal Inference with Experiments
Prediction with Linear Models
Prediction with High-Dim Linear Models
Inference on Causal and Predictive Effects with High-Dim Linear Models
Causal Identification in Observational Data
Potential Outcomes and Conditional Ignorability
Structural Equation Models and Conditional Exogeneity
Directed Acyclic Graphs
Non-Linear Predictive Models and Statistical Inference
Prediction with Non-Linear Models
Inference on Causal Effects with Non-Linear Models
Un-observed Confounding and Instruments
Estimation of Heterogeneous Causal Effects
Identification of Causal Effects in Longitudinal Data
Topics


Linear Predictive Models and Statistical Inference
Causal Inference with Experiments
Prediction with Linear Models
Prediction with High-Dim Linear Models
Inference on Causal and Predictive Effects with High-Dim Linear Models
Causal Identification in Observational Data
Prediction with Non-Linear Models
Directed Acyclic Graphs
Structural Equation Models and Conditional Exogeneity
Potential Outcomes and Conditional Ignorability
Non-Linear Predictive Models and Statistical Inference
Inference on Causal Effects with Non-Linear Models
Un-observed Confounding and Instruments
Estimation of Heterogeneous Causal Effects
Identification of Causal Effects in Longitudinal Data
Topics

Goals for Today

- Meta-Learners for Heterogeneous Treatment Effects
- Neural Network Approaches for Heterogeneous Effects
- Out-of-sample validation and testing

Causal Inference Pipeline



Personalized (Refined) Policies

- To understand who to treat, we need to learn how effect varies
- Conditional Average Treatment Effect

$$\theta(x) = E[\,Y(1) - Y(0) \mid X = x\,]$$

- Allows us to understand differences (heterogeneities) in the response to treatment for different parts of the population
- We can deploy more refined “personalized” policies
- For every person that comes, we observe an $\mathrm{X} = x$ and decide

$$\text{treat if } \theta(x) > 0 \text{ else don't treat}$$

The intrinsic hardness of CATE

- Estimating CATE at least as hard as estimating the best prediction rule
- Inherently harder than estimating an "average"
- So far for our target causal quantities we wanted fast estimation rates and confidence intervals
- We were only ok with "decent" estimation rates for the auxiliary (nuisance) predictive models that entered our analysis

- We might want to relax our goals…

Different Approaches to Relaxing our Goals

- Goal 1: Maybe estimate a simpler projection (e.g. analogue of BLP)
- Goal 2: Confidence intervals for predictions of this simple projection
- Goal 3: Simultaneous confidence bands for predictions of this simple projection
- Goal 4: Estimation error rate for the true CATE
- Goal 5: Confidence intervals for the prediction of a CATE model
- Goal 6: Simultaneous confidence bands for joint predictions of CATE model

Linear Doubly Robust Learner

Meta-learner approaches: S-Learner, T-Learner, X-Learner, R-Learner, DR-Learner
Neural Network approaches: TARNet, CFR
Random Forest approaches: BART

Modified (honest) ML methods: Generalized Random Forest, Orthogonal Random Forest, Sub-sampled Nearest Neighbor Regression

?? (only classical non-parametric statistic results on confidence bands of non-parametric functions)

Policy Learning

- Goal 7: Go after optimal simple treatment policies; give me a policy with value close to the best
- Goal 8: Inference on value of candidate treatment policies
- Goal 9: Inference on value of optimal policy
- Goal 10: Identify responder or heterogeneous sub-groups; policies with statistical significance;

Doubly Robust Policy Evaluation

Doubly Robust Policy Learning

Different Approaches to Relaxing our Goals

- Goal 1: Maybe estimate a simpler projection (e.g. analogue of BLP)
- Goal 2: Confidence intervals for predictions of this simple projection
- Goal 3: Simultaneous confidence bands for predictions of this simple projection
- Goal 4: Estimation error rate for the true CATE
- Goal 5: Confidence intervals for the prediction of a CATE model
- Goal 6: Simultaneous confidence bands for joint predictions of CATE model

Linear Doubly Robust Learner

Meta-learner approaches: S-Learner, T-Learner, X-Learner, R-Learner, DR-Learner
Neural Network approaches: TARNet, CFR
Random Forest approaches: BART

Modified (honest) ML methods: Generalized Random Forest, Orthogonal Random Forest, Sub-sampled Nearest Neighbor Regression

?? (only classical non-parametric statistic results on confidence bands of non-parametric functions)

Policy Learning

- Goal 7: Go after optimal simple treatment policies; give me a policy with value close to the best
- Goal 8: Inference on value of candidate treatment policies
- Goal 9: Inference on value of optimal policy
- Goal 10: Identify responder or heterogeneous sub-groups; policies with statistical significance;

Doubly Robust Policy Evaluation

Doubly Robust Policy Learning

Meta-Learning Approaches for CATE

Meta-Learning Idea

- We assume conditional ignorability: $Y(1), Y(0) \perp\!\!\!\perp D \mid Z$
- We want to estimate the CATE: $E[\,Y(1) - Y(0) \mid X\,], X \subseteq Z$
- If we can frame CATE as a conditional expectation function, then we can deploy any ML approach for solving the corresponding Best Prediction problem

Single Learner (S-Learner)

$$\theta(X) = E[\, g(1, Z) - g(0, Z) \mid X \,], \qquad g(D, Z) = E[Y|D, Z]$$

Meta-Algorithm:

- Run ML regression predicting Y from D, Z to learn g (preferably in a cross-fitting manner, i.e. fit on half the data and predict on the other half and vice versa)
- Run ML regression predicting $g(1, Z) - g(0, Z)$ from X

Two Learner (T-Learner)

$$\theta(X) = E[\, g(1, Z) - g(0, Z) \mid X \,], \qquad g(D, Z) = E[Y|D, Z]$$

Meta-Algorithm:

- Run ML regression predicting Y from Z on subset of data for which $D = 0$ to learn $g(0, \cdot)$ (preferably in a cross-fitting manner)
- Run ML regression predicting Y from Z on subset of data for which $D = 1$ to learn $g(1, \cdot)$ (preferably in a cross-fitting manner)
- Run an ML regression predicting $g(1, Z) - g(0, Z)$ from X

Doubly Robust Learner (DR-Learner)

$$\theta(X) = E[\, Y_{DR}(g,p) \mid X\,], \qquad Y_{DR}(g,p) \coloneqq g(1,Z) - g(0,Z) + H(D,Z)\left(Y - g(D,Z)\right)$$

$$H(D,Z) = \frac{D}{p(Z)} - \frac{1-D}{1-p(Z)}, \qquad g(D,Z) \coloneqq E[Y|D,Z], \qquad p(Z) \coloneqq \Pr(D=1|Z)$$

Meta-Algorithm:

- Run ML regression to estimate $g(1,\ \cdot)$ and $g(0,\ \cdot)$ (either S or T Learner); preferably T-Learner and in cross-fitting manner
- Run ML classification to estimate $\Pr(D=1|Z)$ and calculate $H(D,Z)$; preferably in cross-fitting manner
- Run ML regression predicting $g(1,Z) - g(0,Z) + H(D,Z)\big(Y - g(D,X)\big)$ from X

Cross Learner (X-Learner)

$$\tau(Z) = \tau_1(Z) \coloneqq E[\,Y - E[\,Y \mid D = 0, Z\,] \mid D = 1, Z\,]$$

$$\tau(Z) = \tau_0(Z) \coloneqq E[\,E[\,Y \mid D = 1, Z\,] - Y \mid D = 0, Z\,]$$

For the **control group** I observe $Y(0) \equiv Y(D) = Y$
I can impute a counterfactual outcome $\hat{Y}(1)$, by fitting a response model $\hat{g}_1(Z) \approx E[Y|D = 1, Z]$ from the control group and predict on the treated $\hat{Y}(1) = \hat{g}_1(Z)$

$$Y(1) - Y(0) \mid Z \quad \sim \quad \hat{g}_1(Z) - Y \mid D = 0, Z$$

For the **treated group** I observe $Y(1) \equiv Y(D) = Y$
I can impute a counterfactual outcome $\hat{Y}(0)$, by fitting a response model $\hat{g}_0(Z) \approx E[Y|D = 0, Z]$ from the control group and predict on the treated $\hat{Y}(0) = \hat{g}_0(Z)$

$$Y(1) - Y(0) \mid Z \quad \sim \quad Y - \hat{g}_0(Z) \mid D = 1, Z$$

Cross Learner (X-Learner)

$$\hat{\tau}_1(Z) \coloneqq E[\, Y - \hat{g}_0(Z) \mid D = 1, Z \,]$$

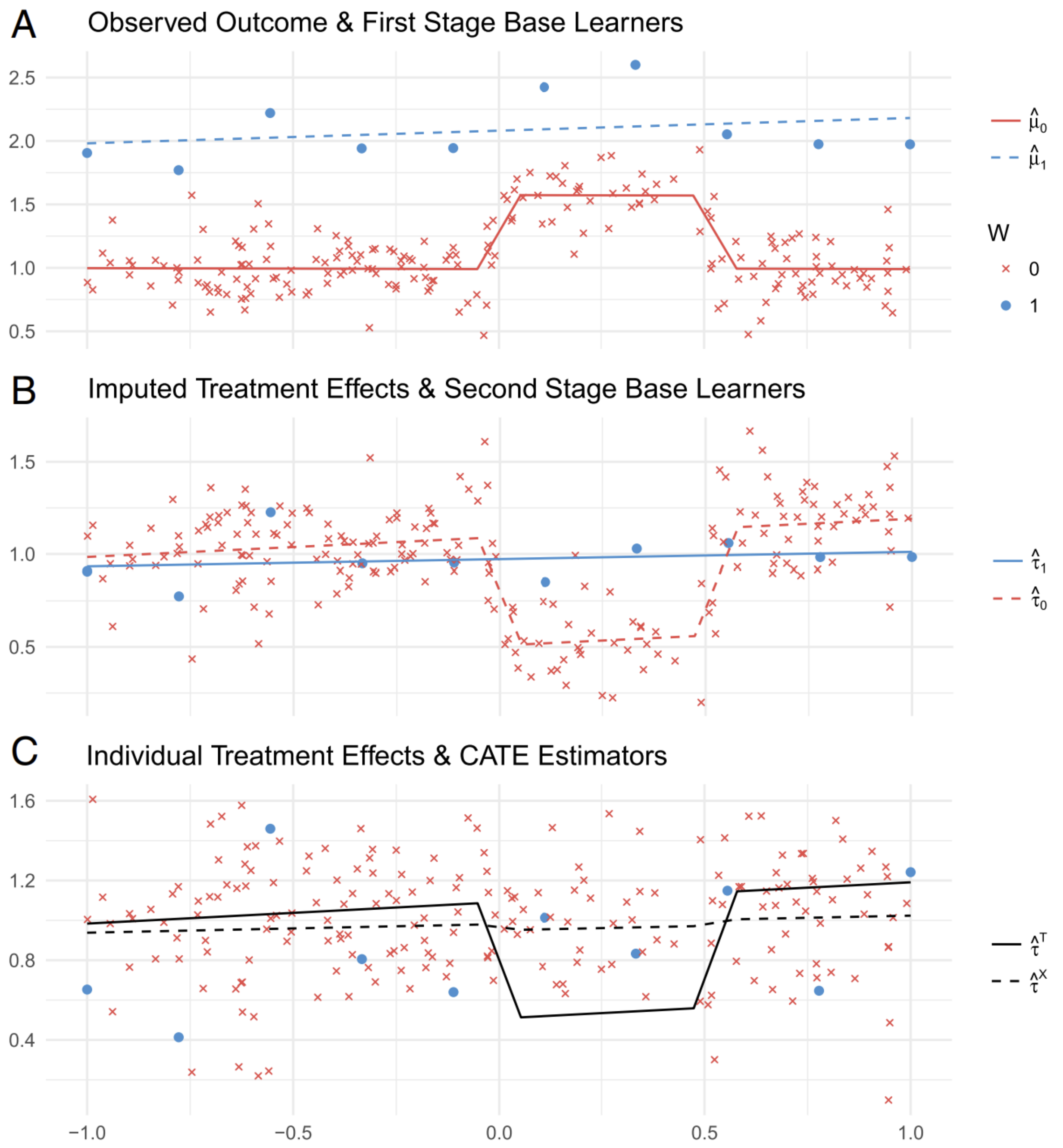
$$\hat{\tau}_0(Z) \coloneqq E[\, \hat{g}_1(Z) - Y \mid D = 0, Z \,]$$

- Which one should we use?
- If for some Z most training data received $D = 1$, then model $\hat{g}_1$ will be a better predictor than $\hat{g}_0$; we should go with $\hat{\tau}_0$
- If for some Z most training data received $D = 0$, then model $\hat{g}_0$ will be a better predictor than $\hat{g}_1$; we should go with $\hat{\tau}_1$

$$\hat{\tau}(Z) = \Pr(D = 1|Z)\, \hat{\tau}_0(Z) + (1 - \Pr(D = 1|Z))\, \hat{\tau}_1(Z)$$

X-Learner

Kunzel et al, 2019



Cross Learner (X-Learner) Meta Algorithm

- Train ML regression $\hat{g}_0$ by predicting Y from Z among control samples
- Construct variables $T_i^1 \coloneqq Y - \hat{g}_0(Z)$ for all treated samples
- Train ML regression $\hat{\tau}_1$ by predicting T_i^1 from Z among treated samples
- Train ML regression $\hat{g}_1$ by predicting Y from Z among treated samples
- Construct variables $T_i^0 \coloneqq \hat{g}_1(Z) - Y$ for all control samples
- Train ML regression $\hat{\tau}_0$ by predicting T_i^0 from Z among control samples
- Train ML classifier to construct $\hat{p}(Z)$ predicting probability $D = 1$ given Z
- Train final ML regression model predicting from X the variable

$$\hat{\tau}(Z) = \hat{p}(Z)\, \hat{\tau}_0(Z) + \left(1 - \hat{p}(Z)\right)\, \hat{\tau}_1(Z)$$

Residual Learner (R-Learner)

- Since we have that:

$$\tau(Z) = E[Y|D = 1, Z] - E[Y|D = 0, Z]$$

- We can write:

$$E[Y|D, Z] = \tau(Z)D + f(Z)$$

- Equivalently:

$$Y = \tau(Z)D + f(Z) + \epsilon, \qquad E[\epsilon|D, Z] = 0$$

- If we further know that $\tau(Z) = \theta(X)$ (effect only depends on X)

$$E[Y|D, Z] = \theta(X)D + f(Z)$$

- We can then write:

$$Y - E[Y|Z] = \theta(X)\,(D - E[D|Z]) + \epsilon$$

Residual Learner (R-Learner)

- If we know that $\tau(Z) = \theta(X)$ (effect only depends on X), we can write
$$\tilde{Y} = \theta(X)\,\widetilde{D} + \epsilon, \qquad E[\epsilon\,|D, Z] = 0$$
- Equivalently, $\theta(\cdot)$ is the minimizer of the square loss:
$$E\left[\left(\tilde{Y} - \theta(X)\widetilde{D}\right)^2\right]$$
- Predict residual outcome $\tilde{Y}$ from residual treatment $\widetilde{D}$ and X with a model of the form $\theta(X)\widetilde{D}$
- Can also be phrased as a "weighted" square loss
$$E\left[\widetilde{D}^2\left(\tilde{Y}/\widetilde{D} - \theta(X)\right)^2\right]$$
- Predict $\tilde{Y}/\widetilde{D}$ from X with sample weights $\widetilde{D}^2$

Residual Learner (R-Learner) Meta Algorithm

- Train ML regression to predict Y from Z and calculate residual $\tilde{Y} \approx Y - E[Y|Z]$ (preferably in cross-fitting manner)
- Train ML regression to predict D from Z and calculate residual $\widetilde{D} \approx D - E[D|Z]$ (preferably in cross-fitting manner)
- Train ML regression with sample weights, to predict $\tilde{Y}/\widetilde{D}$ from X with sample weights $\widetilde{D}^2$

Residual Learner (R-Learner)

- When $\theta(X) = \alpha' \phi(X)$ for some known feature map ϕ then this is equivalent to learning heterogeneous effects with interactions

$$E\left[\left(\tilde{Y} - \alpha' \phi(X) \widetilde{D}\right)^2\right]$$

- Equivalent to OLS with outcome $\tilde{Y}$ and regressors $\phi(X) \widetilde{D}$

Residual Learner (R-Learner)

- If τ does not only depend on X then θ is a "projection"
- But it is a weighted one, it is the minimizer of the loss

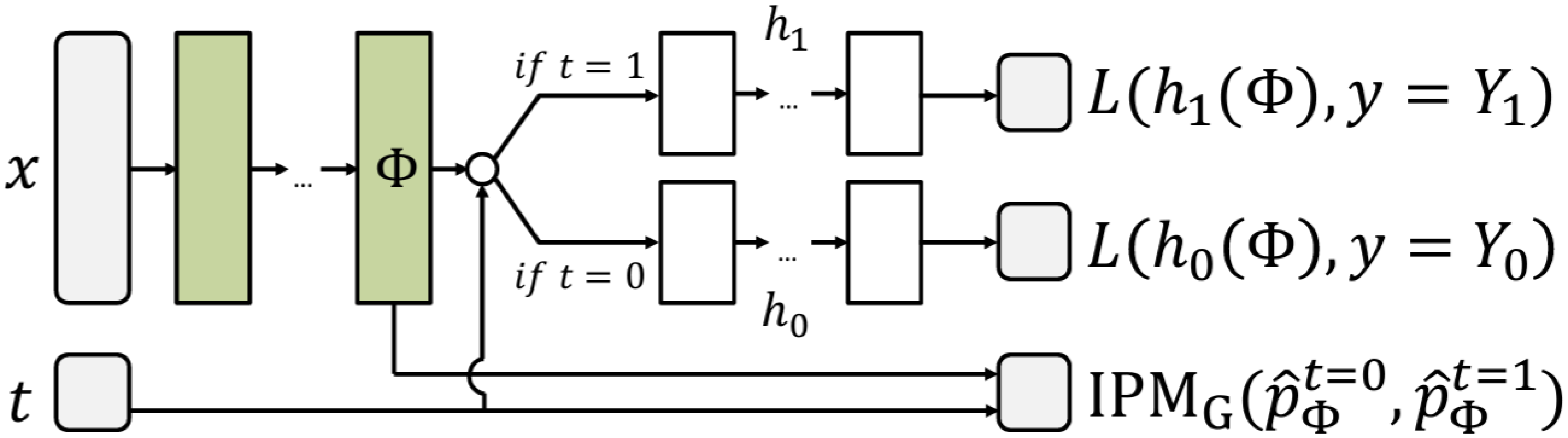
$$E\left[\left(E\left[\tilde{Y} \mid Z, D\right] - \theta(X)\widetilde{D}\right)^2\right] = E\left[\left(\tau(Z)\widetilde{D} - \theta(X)\widetilde{D}\right)^2\right]$$
$$= E\left[\left(\tau(Z) - \theta(X)\right)^2 E\left[\widetilde{D}^2 \mid Z\right]\right] = E\left[\left(\tau(Z) - \theta(X)\right)^2 Var(D|Z)\right]$$

- We put more weight on regions of Z with more randomized treatment
- If some regions of the population were assigned treatments roughly deterministically, then they are ignored in the approximation

Comparing Meta-Learners

- S and T-Learners are typically poor performing as they heavily depend on outcome modelling; among them the T-Learner should be preferred
- X-Learner is a better version of S and T as it incorporates propensity knowledge
- DR-Learner and R-Learner, both possess “Neyman orthogonality” properties as they carefully combine outcome and treatment assignment modelling
- The error of the final cate model is not heavily impacted by the errors in the auxiliary models (Orthogonal Statistical Learning)
- DR-Learner estimates un-weighted projection of true CATE on model space, but can be “high-variance” due to inverse propensity
- R-Learner estimates variance weighted projection but is much more stable to extreme propensities as it never divides by propensity.

Neural Network CATE Learners (CFR Net)
Shalit et al. 17



Model Selection and Evaluation

Model Selection and Evaluation

- Each of the meta learners is defined based on a loss function
- We can use loss function for model selection and out-of-sample performance evaluation
- To compare across any CATE learner, we can evaluate based on an "orthogonal loss", which is robust to nuisance estimation
- **R-Loss:** for a separate sample, calculate residuals $\tilde{Y}, \widetilde{D}$ in a cross-fitting manner. For any candidate CATE model θ evaluate
$$L(\theta) \coloneqq E\left[\left(\tilde{Y} - \theta(X)\widetilde{D}\right)^2\right]$$
- **DR-Loss:** for a separate sample, calculate regression model g (using T-Learner) and propensity model p. For any candidate CATE model θ evaluate
$$L(\theta) \coloneqq E\left[\left(Y_{DR}(g, p) - \theta(X)\right)^2\right]$$

Evaluation via Testing Approaches

- If CATE model θ was good, then BLP of CATE, when using $\left(1, \theta(X)\right)$ as the feature map, should assign a lot of weight on $\theta(X)$

- Run OLS regression predicting $Y_{DR}(g, p)$ using regressors $\left(1, \theta(X)\right)$

$$E\left[\left(Y_{DR}(g, p) - \beta_0 - \beta_1 \theta(X)\right)^2\right]$$

- Construct confidence intervals and test whether $\beta_1 \neq 0$; then $\theta(X)$ correlates with the true CATE! Ideally $(\beta_0 = 0, \beta_1 = 1)$

Validation via GATEs

- For any large enough group G, we can calculate out-of-sample group average effects by simply averaging $Y_{DR}(g, p)$
$$GATE(G) \coloneqq E[Y(1) - Y(0) | X \in G] = E[Y_{DR}(g, p) | X \in G]$$

- If the CATE model θ is accurate, then if we restrict to some group G then the average of θ over this group, should match the out-of-sample group average treatment effect
$$E[\theta(X) | X \in G] \approx GATE(G)$$

- We can measure such GATE discrepancies out-of-sample

Validation via Calibration

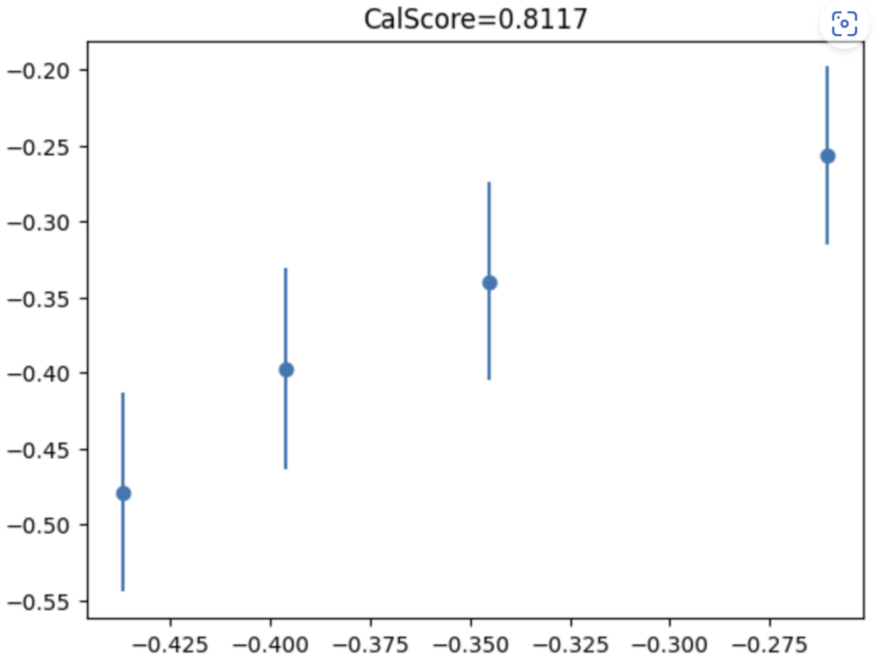
- One natural definition of groups is the “percentile groups of the CATE predictions”
- For the top 25% of the CATE predictions based on the model θ, the mean of model predictions, should match the out-of-sample GATE for that group
- Consider a set of quantiles $q_1, .., q_K$ (e.g. 0, 25, 50, 75)
- Consider the distribution D of $\theta(X)$ over the training data X
- Let G_i be the groups defined as $\{X: \theta(X) \in [q_i\ q_{i+1}]\ quantile\ of\ D\}$

$$\tau_i := E[\theta(X)|X \in G_i] \approx GATE(G_i) := E[Y_{DR}(g,p)|X \in G_i]$$

- Calibration score:

$$\mathrm{CalScore}(\theta) := \sum_i \Pr(G_i)\,|\tau_i - GATE(G_i)|$$

- Normalized calibration score: $1 - \frac{\mathrm{CalScore}(\theta)}{\mathrm{CalScore}(constant\ CATE = E[Y_{DR}(g,p)])}$



Testing for Heterogeneity

- We can easily construct joint confidence intervals for all the GATEs
- GATEs are the coefficients in the BLP of CATE using group one-hot-encoding as features

$$E\left[\left(Y_{DR}(g,p) - \beta'(1\{X \in G_1\}, \dots, 1\{X \in G_K\})\right)^2\right]$$

- We can use joint confidence intervals for BLP via the DR-Learner
- If there was heterogeneity, then we should have that there are GATEs whose confidence intervals are non-overlapping

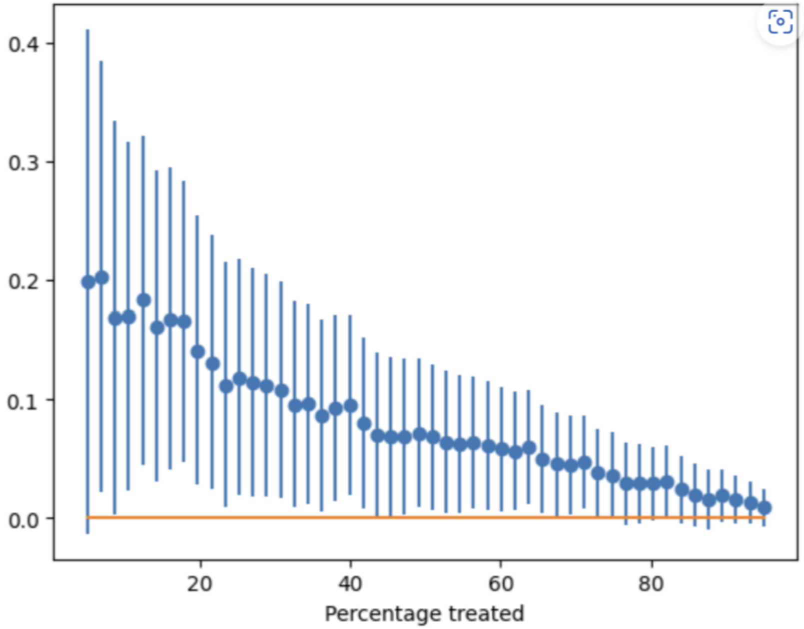
Stratification Motivated Evaluation

- If we were to "prioritize" into treatment based on θ with a target to treat around q-percent of the population then what would be the ATE of the treated group
- Consider the distribution D of $\theta(X)$ over the training data X
- We can define the group:

$$G_q := \{X: \theta(X) \geq (1-q) - th\ quantile\ of\ D\}$$

$$\tau(q) = E\left[\, Y_{DR}(g,p) \mid X \in G_q \,\right] - E[Y_{DR}(g,p)]$$

- Ideally, $\tau(q)$ should be always positive and increasing!
- AUTOC $\approx$ the area under the curve $\tau(q)$



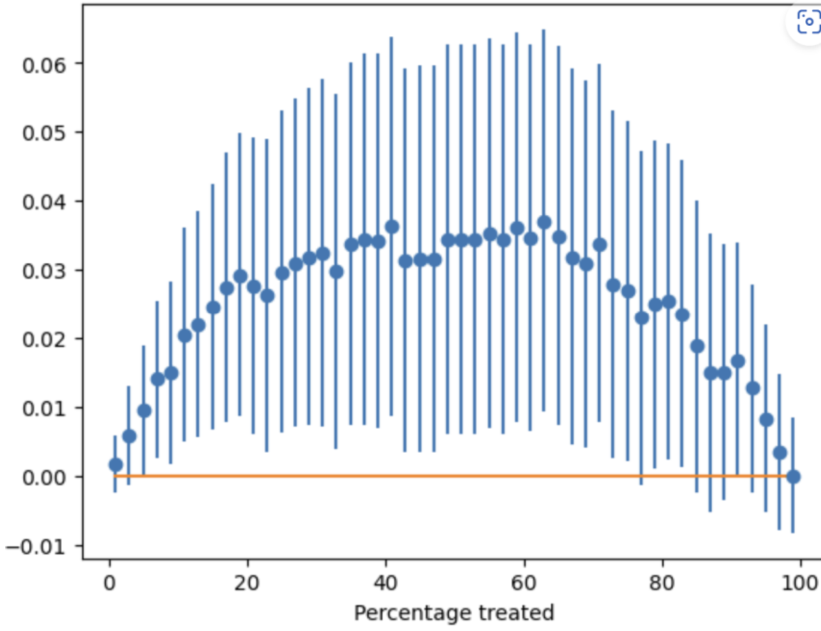
Stratification Motivated Evaluation

- If we were to "prioritize" into treatment based on θ with a target to treat around q-percent of the population then what would be the policy value we would get over treating q percentage at random
- Consider the distribution D of $\theta(X)$ over the training data X
- We can define the group:

$$G_q \coloneqq \{X\colon \theta(X) \geq (1-q) - th\ quantile\ of\ D\}$$

$$\tau(q) = q\left(E\left[\, Y_{DR}(g,p) \,\middle|\, X \in G_q \,\right] - E[Y_{DR}(g,p)]\right)$$

- Ideally, $\tau(q)$ should be large positive for some values!
- QINI: the area under the curve $\tau(q)$



Policy Learning

Candidate Policy

- What if I have a candidate policy π on who to treat
- The average policy effect is of the form:
$$V(\pi) = E\left[\pi(X)\left(Y(1) - Y(0)\right)\right]$$
- Under conditional ignorability:
$$V(\pi) = E[\pi(X)(E[Y|D = 1, Z] - E[Y|D = 0, Z])]$$
- We can also measure performance via the doubly robust outcome
$$V(\pi) = E[\pi(X)\, Y_{DR}(g, p)]$$
- Also falls in the Neyman orthogonal moment estimation framework
$$E[\pi(X) Y_{DR}(g, p) - \theta] = 0$$
- We can easily construct confidence intervals

Policy Optimization

- We can optimize over a space of policies Π on the samples

$$\hat{V}(\pi) = E_n[\pi(X)Y_{DR}(\hat{g}, \hat{p})]$$

- Regret:

$$\max_{\pi \in \Pi} V(\pi) - V(\hat{\pi})$$

- Regret not impacted a lot by errors in $\hat{g}$ or $\hat{p}$
- Performance as if true g, p were known (assuming estimation rates are $n^{-\frac{1}{4}}$

Non-Parametric Confidence Intervals

Generalized Random Forest

- We want to estimate a solution to a conditional moment restriction
$$\theta(x) \coloneqq E[\, m(Z;\theta) \mid X = x\,]$$
- We do so by splitting constructing a tree that at each level optimizes the heterogeneity of the values of the local solution created at the resulting children nodes
- At the end we have many trees each defining a neighborhood structure
- For every candidate x we use the trees to define a set of weights with every training point and we solve the moment equation
$$\sum_i w_i(x)\, m(Z_i;\theta) = 0$$

Generalized Random Forest

- If each tree is built in an honest manner (i.e. samples used in the final weighted moment equation are separate from samples used to determine splits)
- If each tree is build in a balanced manner (at least some constant fraction on each side of the split)
- If each tree is built on a sub-sample without replacement, of an appropriate size
- Then the prediction $\theta(x)$ is asymptotically normal and we can construct confidence intervals via an appropriate bootstrap procedure

GRF for CATE

- We can do this with the residual moment:

$$E\left[\left(\tilde{Y} - \theta(x)\widetilde{D}\right)\widetilde{D} \mid X = x\right] = 0$$

- (Orthogonal Random Forest) We can also do a similar approach with the doubly robust targets

$$E[\, Y_{DR}(g, p) - \theta(x) \mid X = x \,] = 0$$

- We can also do this even when X is a subset of Z