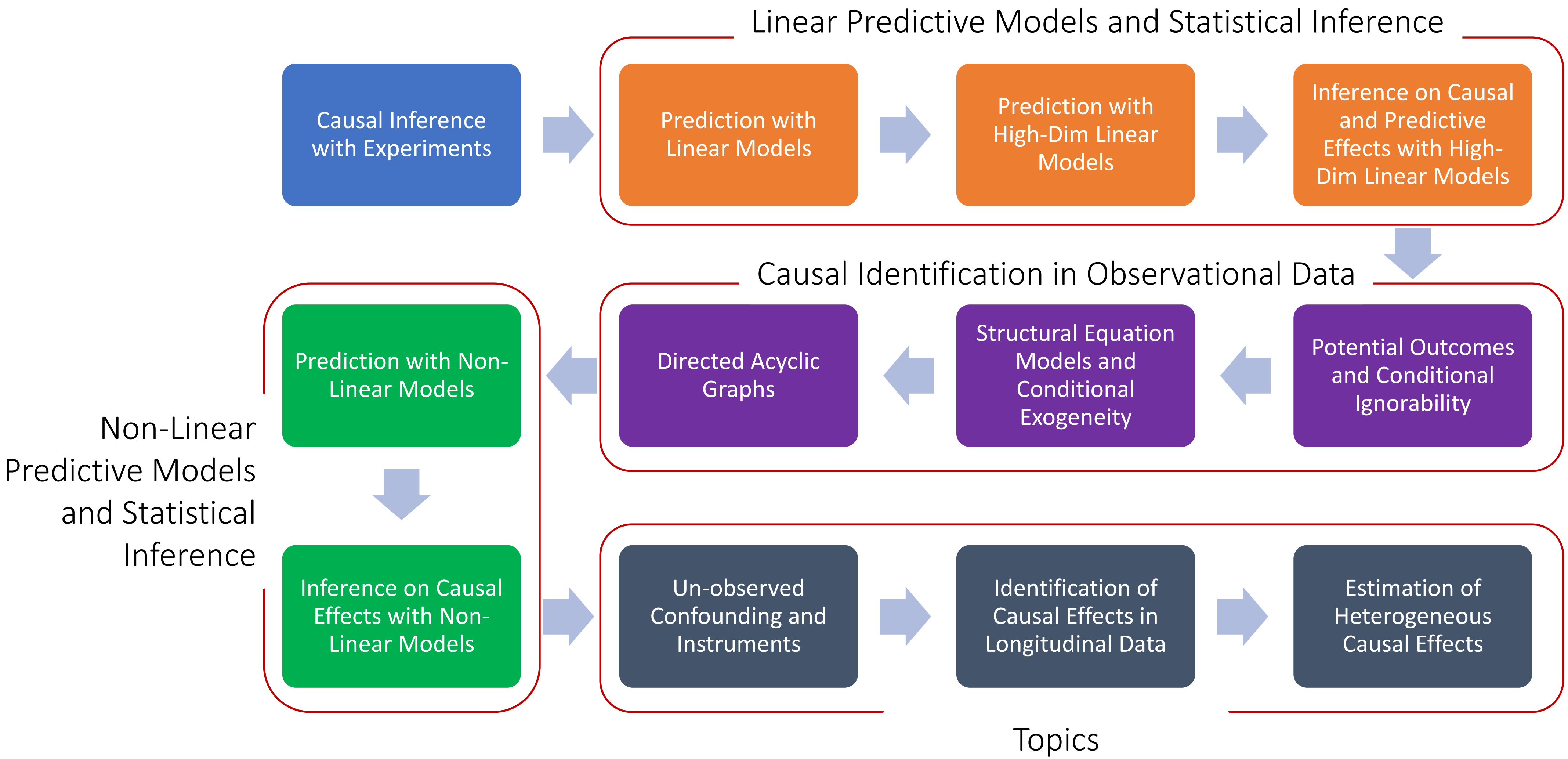
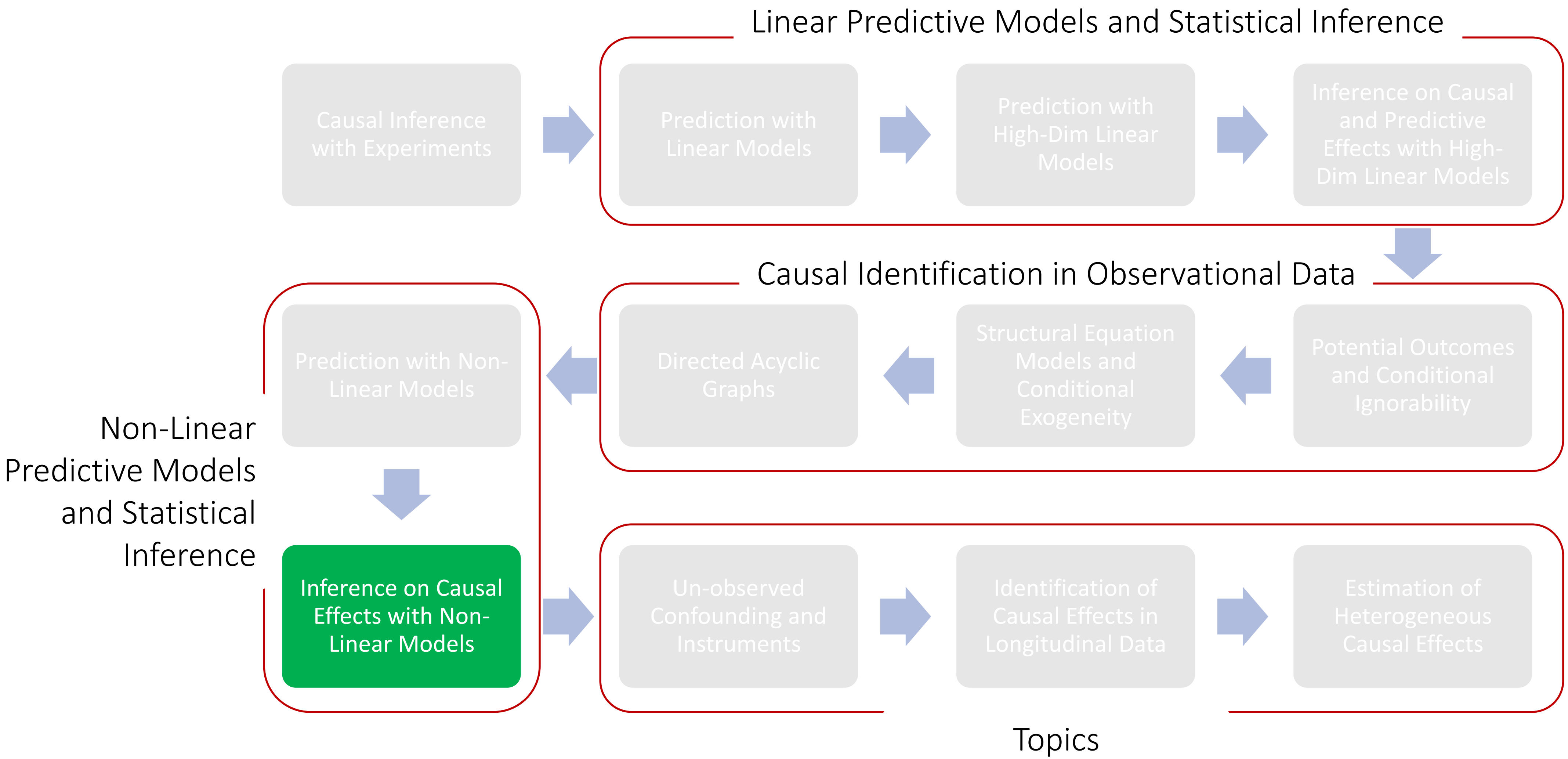
MS&E 228: Inference with Modern Non-Linear Prediction

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
Prediction with Non-Linear Models
Non-Linear Predictive Models and Statistical Inference
Inference on Causal Effects with Non-Linear Models
Un-observed Confounding and Instruments
Identification of Causal Effects in Longitudinal Data
Estimation of Heterogeneous Causal Effects
Topics

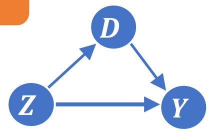

Linear Predictive Models and Statistical Inference
Causal Inference with Experiments
Prediction with Linear Models
Prediction with High-Dim Linear Models
Inference on Causal and Predictive Effects with High-Dim Linear Models
Causal Identification in Observational Data
Potential Outcomes and Conditional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

Recap of Last Lecture

Causal Inference Pipeline

Theory

DAGs/ASEMs



$$a_d = E[E[Y|D=d,Z]]$$

Construct estimate $\hat{g}(d,z)$ of predictive model

$$E[\,Y \mid D=d, Z=z\,]$$

Estimate mean counterfactual

$$\hat{a}_d = E_n[\hat{g}(d,Z)]$$

$$a_d \in \hat{a}_d \pm 2\hat{\sigma}$$

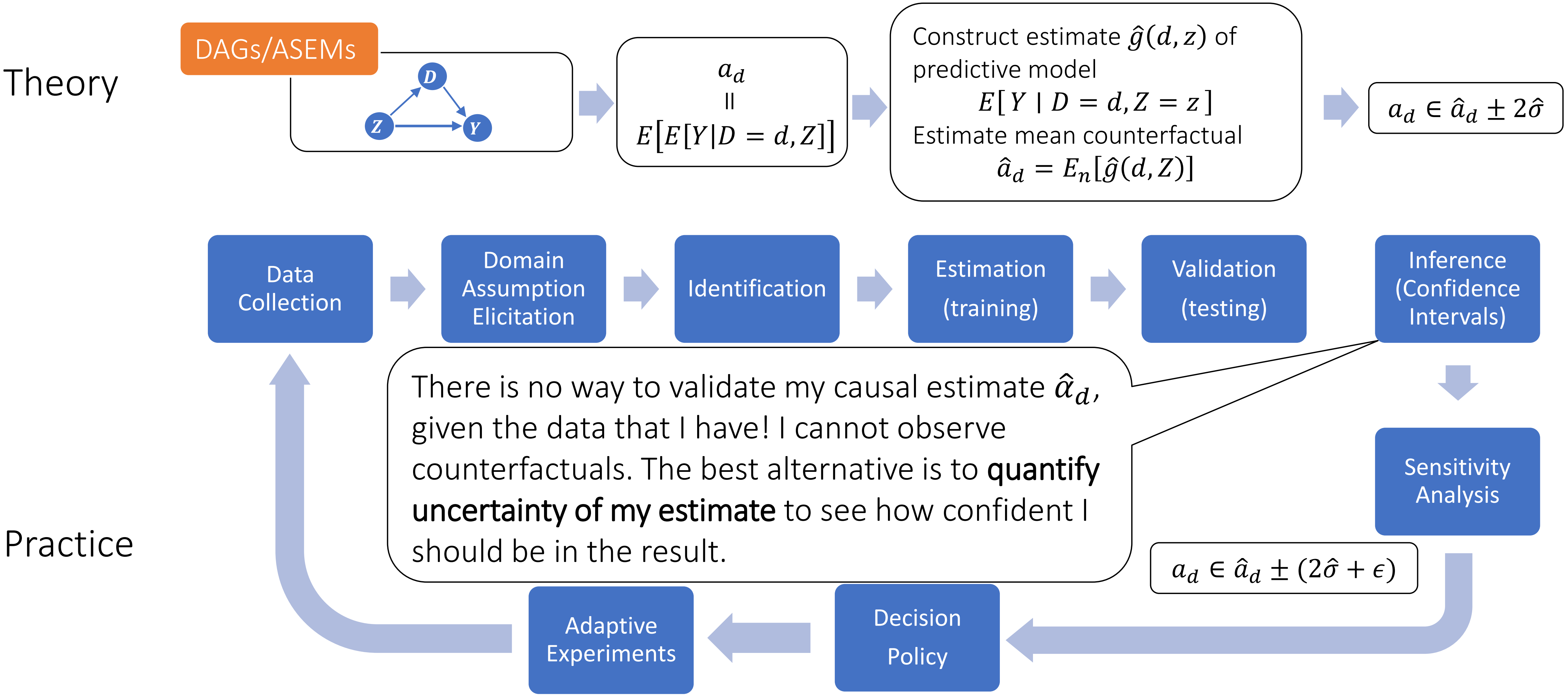
Practice

Data Collection → Domain Assumption Elicitation → Identification → Estimation (training) → Validation (testing) → Inference (Confidence Intervals) → Sensitivity Analysis → Decision Policy → Adaptive Experiments → Data Collection

So I need to estimate this **bigger predictive model, predicting the outcome from the treatment $\boldsymbol{D}$ and the controls $\boldsymbol{S}$,** in some manner. Then average the predictive values over S fixing $d = 1$.

$$a_d \in \hat{a}_d \pm (2\hat{\sigma} + \epsilon)$$

Causal Inference Pipeline



Goals for Today

- Methods for Confidence Intervals for ATE with non-linear models
- General Neyman Orthogonality Framework (Double/Debiased ML)
- Methods for Confidence Intervals for ATE in a partially-linear model
- Sample-splitting and cross-fitting

- Proof sketch of main theorem*

The Example Problem

Identification under Conditional Ignorability

- Once we condition on enough variables X that affect treatment assignment, remnant variation in D is exogenous (as-if trial)

$$Y^{(d)} \perp D \mid X \quad \text{(conditional ignorability)}$$

- Why useful:

$$\begin{aligned} E[Y \mid D = d, X] &= E\left[Y^{(D)} \mid D = d, X\right] \\ &= E\left[Y^{(d)} \mid D = d, X\right] = E\left[Y^{(d)} \mid X\right] \end{aligned}$$

- Average treatment effect is “identified” as (g-formula):

$$\begin{aligned} \theta_0 = E\left[Y^{(1)} - Y^{(0)}\right] &= E\left[E\left[Y^{(1)} - Y^{(0)} \mid X\right]\right] \\ &= E\left[E[Y|D = 1, X] - E[Y \mid D = 0, X]\right] \end{aligned}$$

Let’s take it to data

- We observe n samples $Z_1, \dots, Z_n$ where $Z_i = (X_i, D_i, Y_i)$

- Want to estimate average effect θ_0, which satisfies:
$$\theta_0 = E[g_0(1, X) - g_0(0, X)]$$
- Where:
$$g_0(D, X) \coloneqq E[\,Y \mid D, X\,]$$

- We want to be able to use ML to learn regression functiong_0!

A General Estimation Framework

Semi-Parametric Moment Restrictions

- Observe samples $Z_1, \ldots, Z_n$ i.i.d. from data distribution D

- Distribution D satisfies vector of moment restrictions
$$M(\theta_0, g_0) \coloneqq E_{Z\sim D}[m(Z; \theta_0, g_0)] = 0$$

- $\theta_0 \in R^d$ finite dimensional target parameter of interest
- $g_0 \in G$ potentially infinite dimensional (an un-known function) we don't care (nuisance)
- g_0 is un-known and needs to be estimated from data

ATE under Conditional Exogeneity

- We observe n samples $Z_1, \ldots, Z_n$ where $Z_i = (X_i, D_i, Y_i)$

- Want to estimate average effect θ_0, which satisfies:

$$M(\theta_0, g_0) \coloneqq E[g_0(1, X) - g_0(0, X) - \theta_0] = 0$$

- Where:

$$g_0(D, X) \coloneqq E[\, Y \mid D, X \,]$$

- We want to be able to use ML to learn regression function g!

Given n samples we want to produce estimate $\hat{\theta}$

What do we want from $\hat{\theta}$?

- Consistency

$$\hat{\theta} \to_p \theta_0$$

What do we want from $\hat{\theta}$?

- Finite sample parametric rate:

$$\left\|\hat{\theta} - \theta_0\right\| = O_p\left(\sqrt{d/n}\right)$$

What do we want from $\hat{\theta}$?

- Asymptotic normality

$$\sqrt{n}\left(\hat{\theta} - \theta_0\right) \to_d N\left(0, \sigma^2\right)$$

- Construction of confidence intervals:

with prob. $\approx$ 95%: $\theta_0 \in \left[\hat{\theta} \pm 1.96\hat{\sigma}/\sqrt{n}\,\right]$.

- Calculation of p-value for zero effect

What do we want from $\hat{\theta}$?

- Asymptotic linearity

$$\sqrt{n}\left(\hat{\theta} - \theta_0\right) = \frac{1}{\sqrt{n}}\sum_{i=1}^{n} \phi_0(Z_i) + o_p(1)$$

- Consistency of bootstrap confidence intervals

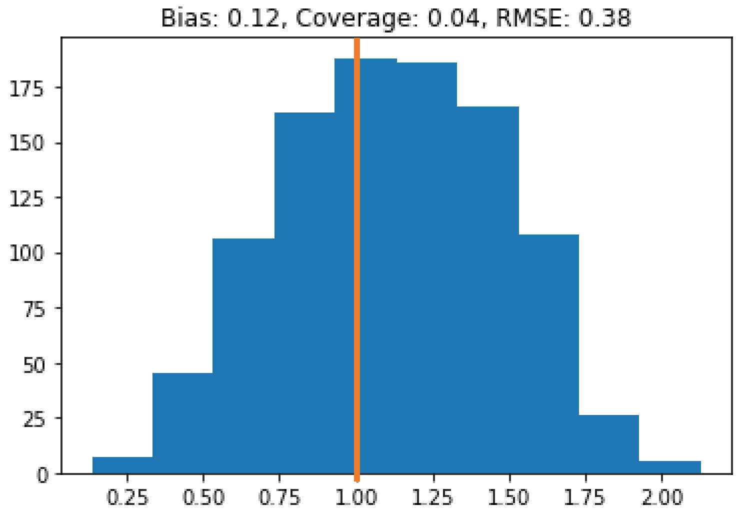
Natural Estimation Algorithm

- Estimate $\hat{g}$ of g_0 from data
- Return solution $\hat{\theta}$ to empirical plug-in moment equation:

$$M_n(\hat{\theta}, \hat{g}) \coloneqq \frac{1}{n}\sum_{i=1}^{n} m(Z_i; \hat{\theta}, \hat{g}) = 0$$

Natural Algorithm Gone Wrong

```
def est(X, D, y): # direct non-orthogonal estimator of average effect
    est = RandomForestRegressor(min_samples_leaf=20)
    est.fit(np.hstack([D.reshape(-1, 1), X]), y)
    ones = np.hstack([np.ones((X.shape[0], 1)), X])
    zeros = np.hstack([np.zeros((X.shape[0], 1)), X])
    preds = est.predict(ones) - est.predict(zeros)
    return np.mean(preds), np.std(preds)/np.sqrt(X.shape[0])
```



Natural Estimation Algorithm (Draft 2)

- Split the data in half
- On first half, estimate $\hat{g}$ of g_0
- On second half, solution $\hat{\theta}$ to empirical plug-in moment equation:

$$M_n(\hat{\theta}, \hat{g}) \coloneqq \frac{1}{n_2} \sum_{i \in S_2} m(Z_i; \hat{\theta}, \hat{g}) = 0$$

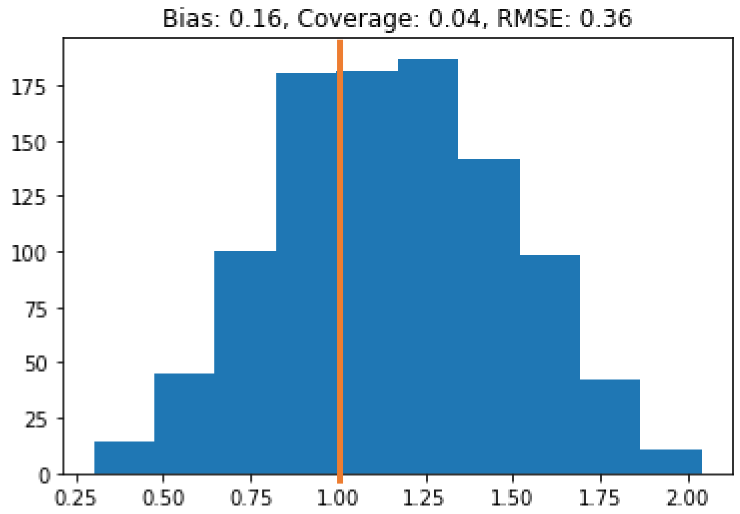
Natural Estimation Algorithm (Draft 3)

- Split data in K parts, $S_1, \ldots, S_K$
- For each part k, estimate $\hat{g}_k$ using data from all parts except S_k
- Return solution $\hat{\theta}$ to cross-fitted empirical moment equation:

$$\frac{1}{n} \sum_{k=1}^{K} \sum_{i \in S_k} m\left(Z_i; \hat{\theta}, \hat{g}_k\right) = 0$$

Natural Algorithm (Draft 3) Gone Wrong

```python
def est2(X, D, y): # direct non-orthogonal estimator with sample splitting
    effects = np.zeros(X.shape[0])
    for train, test in KFold(n_splits=3).split(X):
        est = RandomForestRegressor(min_samples_leaf=20)
        est.fit(np.hstack([D[train].reshape(-1, 1), X[train]]), y[train])
        ones = np.hstack([np.ones((X[test].shape[0], 1)), X[test]])
        zeros = np.hstack([np.zeros((X[test].shape[0], 1)), X[test]])
        effects[test] = est.predict(ones) - est.predict(zeros)
    return np.mean(effects), np.std(effects)/np.sqrt(X.shape[0])
```

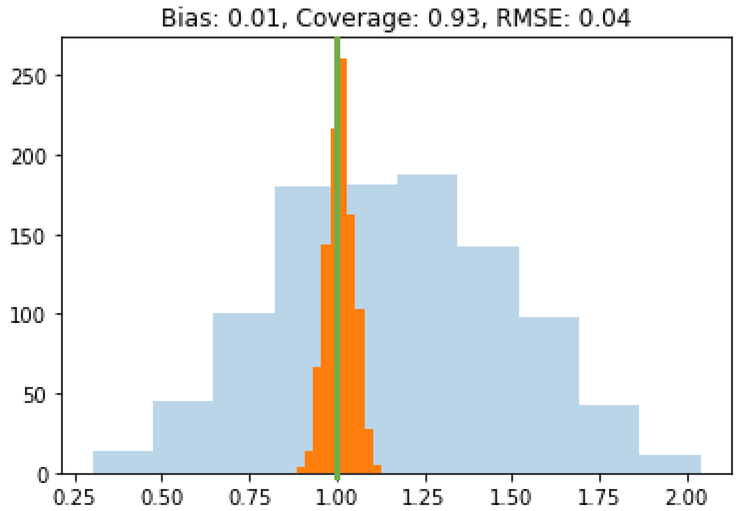


When is estimate $\hat{\theta}$ $\sqrt{n}$-asymptotically normal?

Natural Algorithm (Draft 3) Gone Right

[redacted]

TO BE UNCOVERED!



When is estimate $\hat{\theta}$ $\sqrt{n}$-asymptotically normal?
We need to change the moment we use

Debiasing Intuition

ATE under Conditional Exogeneity

- We observe n samples $Z_1, \ldots, Z_n$ where $Z_i = (X_i, D_i, Y_i)$
- Want to estimate average effect θ_0, which satisfies:

$$M(\theta_0, g_0) \coloneqq E[g_0(1, X) - g_0(0, X) - \theta_0] = 0$$

- Where:

$$g_0(D, X) \coloneqq E[\, Y \mid D, X \,]$$

- The moment $M(\theta, g)$ is sensitive to variations in g
- Any bias or error in g propagates to bias or error in moment and $\hat{\theta}$
- Can we add a correction that corrects the biases of $\hat{g}$

Better Moment for ATE

- Add a "debiasing" correction:

$$\widetilde{M}(\theta, g, a) = M(\theta, g) + E\big[a(D, X)\,\big(Y\; - g(D, X)\big)\big]$$

- What is a_0? Should be such that

$$E[a_0(D, X)\; g(D, X)] = E[g(1, X) - g(0, X)]$$

- If this holds then if g is very wrong but a is correct:

$$\theta = E[a(D, X)Y] = E\big[a(D, X)E[\,Y \mid D, X\,]\big]$$
$$= E[a(D, X)g(D, X)] = E[g(1, X) - g(0, X)]$$

Inverse Propensity Weighting (IPW)

- The following works: inverse propensity scoring

$$a_0(D,X) = \frac{D}{\Pr[D=1|X]} - \frac{1-D}{\Pr[D=0|X]}$$

- Sketch:

$$\begin{aligned} E\left[\frac{D}{\Pr[D=1|X]}g(D,X)\right] &= E\left[\frac{D}{\Pr[D=1|X]}g(1,X)\right] \\ &= E\left[\frac{E[D|X]}{\Pr[D=1|X]}g(1,X)\right] \\ &= E[g(1,X)] \end{aligned}$$

New Moment is Insensitive

$$\widetilde{M}(\theta, g, a) = M(\theta, g) + E\big[a(D, X)\,\big(Y\ - g(D, X)\big)\big]$$

- Take derivative with respect to g at θ_0, g_0, a_0 in any direction $\nu \in G$

$$\frac{\partial}{\partial t}\widetilde{M}(\theta_0, g_0 + t\,\nu, a_0)\bigg|_{t=0} = E[\nu(1, X) -\ \nu(0, X)] - E[a(D, X)\,\nu(D, X)]$$
$$= 0$$

- Take derivative with respect to a at θ_0, g_0, a_0 in any direction $\nu \in A$

$$\frac{\partial}{\partial t}\widetilde{M}(\theta_0, g_0, a_0 + t\nu)\bigg|_{t=0} = E\big[\nu(D, X)\,\big(Y - g_0(D, X)\big)\,\big]$$
$$= 0$$

Neyman Orthogonality

Formal Definition

- Moment $M(\theta, g)$ is Neyman orthogonal if for any $\nu \in G - g_0$:

$$D_g M(\theta_0, g_0)[\nu] \coloneqq \frac{\partial}{\partial t} M(\theta_0, g_0 + t\,\nu)\Big|_{t=0} = 0$$

Sample-Splitting Estimation Algorithm

- Split the data in half
- On first half, estimate $\hat{g}$ of g_0
- On second half, solution $\hat{\theta}$ to empirical plug-in moment equation:

$$M_n(\hat{\theta}, \hat{g}) := \frac{1}{n_2} \sum_{i \in S_2} m(Z_i; \hat{\theta}, \hat{g}) = 0$$

Main Theorem

- If moment is Neyman orthogonal and RMSE of $\hat{g}$ is $o_p\left(n^{-1/4}\right)*$

$$\sqrt{n}\left(\hat{\theta}-\theta_0\right) \rightarrow N\left(0, A^{-1}\Sigma\left(\mathrm{A}^{-1}\right)^{\top}\right)$$

- $A = \nabla_\theta M(\theta_0, g_0)$ and $\Sigma = E\left[m(Z;\theta_0, g_0)\, m(Z;\theta_0, g_0)^{\top}\right]$

*plug regularity conditions

Continuous Treatments under Partial Linearity

Partially Linear Model

- Relevant in many applications: dose-response curve in healthcare, effect of price on demand, return-on-investment
- Assume conditional exogeneity
$$Y^{(d)} \perp D \mid X$$
- Assume partially linear response
$$g_0(D, X) = E[\,Y \mid D, X\,] = \theta_0 D + f_0(X)$$
- Parameter of interest θ_0 is constant marginal effect of treatment

Partially Linear Model

- By definition of CEF we have the decomposition

$$Y = g_0(D, X) + \epsilon = \theta_0 D + f_0(X) + \epsilon, \qquad E[\epsilon | D, X] = 0$$

- By definition of g_0, for any function $q(D, X)$

$$E\left[\left(Y - \theta_0 D - f_0(X)\right) q(D, X)\right] = E[E[Y - g_0(D, X) \mid D, X]\, q(D, X)] = 0$$

- Direct non-orthogonal method, estimate $\hat{f}$ and solve:

$$E\left[\left(Y - \theta D - \hat{f}(X)\right) D\right] = 0$$

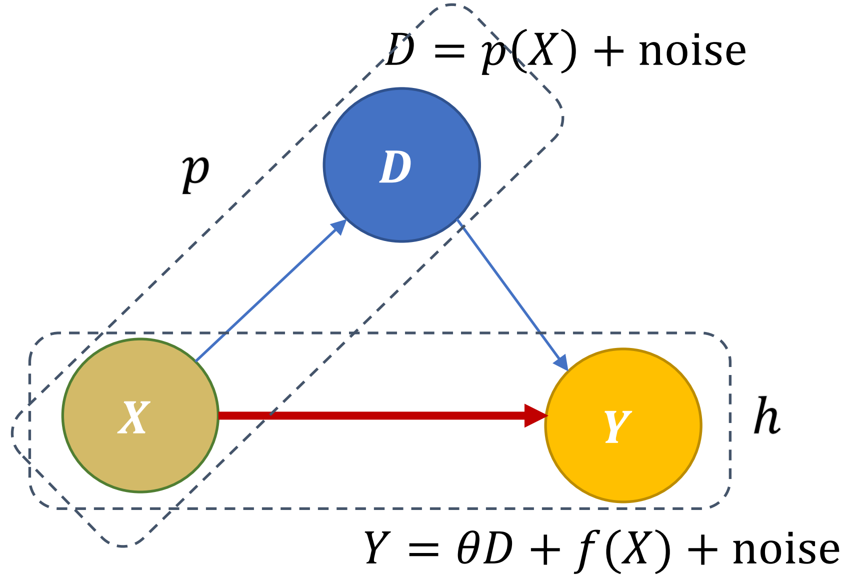
Generalization of FWL Theorem

- Let's define a slight variant of residualization
$$\tilde{V} = V - E[V|X]$$
- Generalization of FWL theorem to partially linear models
$$\tilde{Y} = \theta_0 \widetilde{D} + \epsilon, \qquad E\left[\epsilon \middle| \widetilde{D}\right] = 0$$
- Let's consider the residual outcome

$$\begin{aligned}
\tilde{Y} &= Y - E[Y|X] \\
&= \theta_0 D + f_0(X) + \epsilon - E[\theta_0 D + f_0(X) + \epsilon|X] \\
&= \theta_0 D + f_0(X) + \epsilon - \theta_0 E[D|X] - f_0(X) \\
&= \theta_0 (D - E[D|X]) + \epsilon
\end{aligned}$$

Orthogonal Method: Double ML

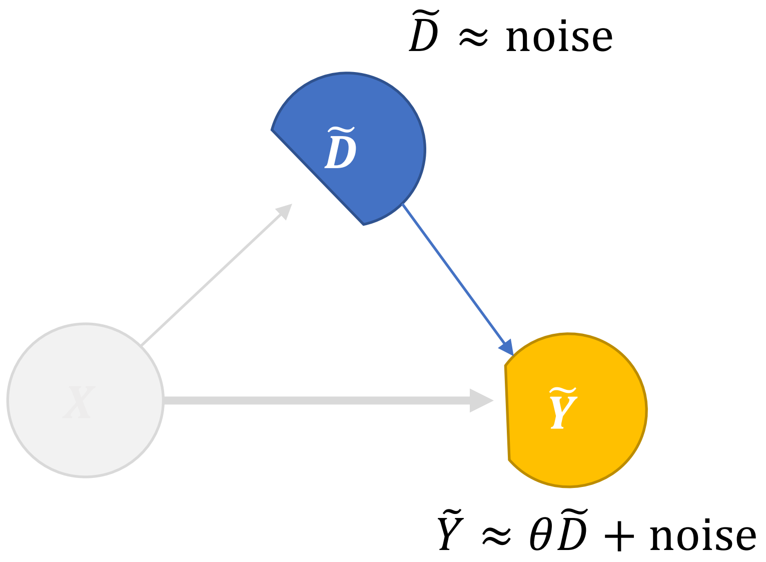
- Double ML. Split samples in half
  - Regress $Y \sim X$ with ML on first half, to get estimate $\hat{h}(S)$ of $E[Y|X]$
  - Regress $D \sim X$ with ML on first half, to get estimate $\hat{p}(S)$ of $E[D|X]$



Orthogonal Method: Double ML

- Double ML. Split samples in half
  - Regress $Y \sim X$ with ML on first half, to get estimate $\hat{h}(S)$ of $E[Y|X]$
  - Regress $D \sim X$ with ML on first half, to get estimate $\hat{p}(S)$ of $E[D|X]$
  - Construct residuals on other half, $\tilde{D} \coloneqq D - \hat{p}(X)$ and $\tilde{Y} \coloneqq Y - \hat{h}(X)$
  - Run OLS on residuals: $\tilde{Y} \sim \tilde{D}$ to get $\hat{\theta}$

- OLS equivalent to solving moment condition:
$$E\left[\left(\tilde{Y} - \theta\tilde{D}\right)\tilde{D}\right] = 0$$
- Orthogonal Moment condition:
$$M(\theta, h, p) = E\left[\left(Y - h(X) - \theta\left(D - p(X)\right)\right)\ \left(D - p(X)\right)\right]$$



Practical Variants of Sample-Splitting

Cross-fitting and semi-cross-fitting

Cross-fitting

- Sample splitting is statistically lossy

- Only half of the data are used for the final parameter estimation
- Can we utilize all the data?

- *Cross-fitting:* analogous to cross-validation
- Use the second half to train g and predict on first half
- Then calculate parameter using all the data

Cross-fitting Estimation Algorithm

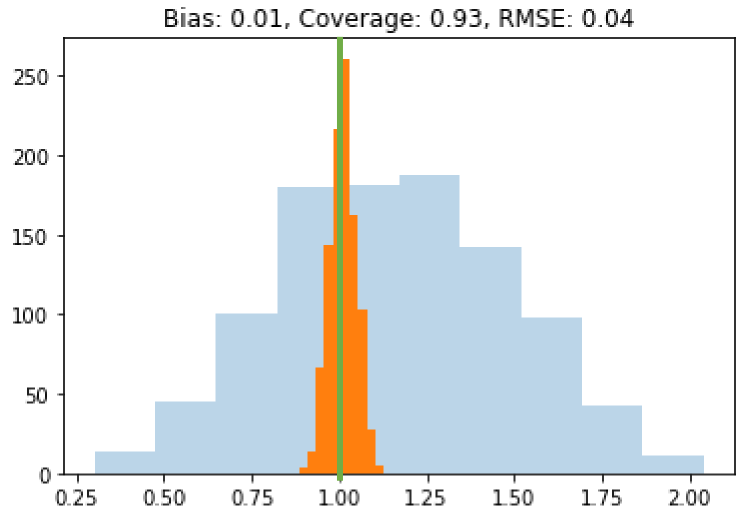
- Split the data in half
- On first half, estimate $\hat{g}_1$ of g_0 and predict on second half
- On second half, estimate $\hat{g}_2$ of g_0 and predict on first half
- On all data, solution $\hat{\theta}$ to empirical plug-in moment equation:

$$M_n(\hat{\theta}, \hat{g}) \coloneqq \frac{1}{n} \sum_{i \in S_1} m(Z_i; \hat{\theta}, \hat{g}_2) + \frac{1}{n} \sum_{i \in S_2} m(Z_i; \hat{\theta}, \hat{g}_1) = 0$$

- In practice do this with $K \approx 3 \text{ to } 5$ folds: for each fold k train on all other folds and predict on fold k

Natural Algorithm (Draft 3) Gone Right

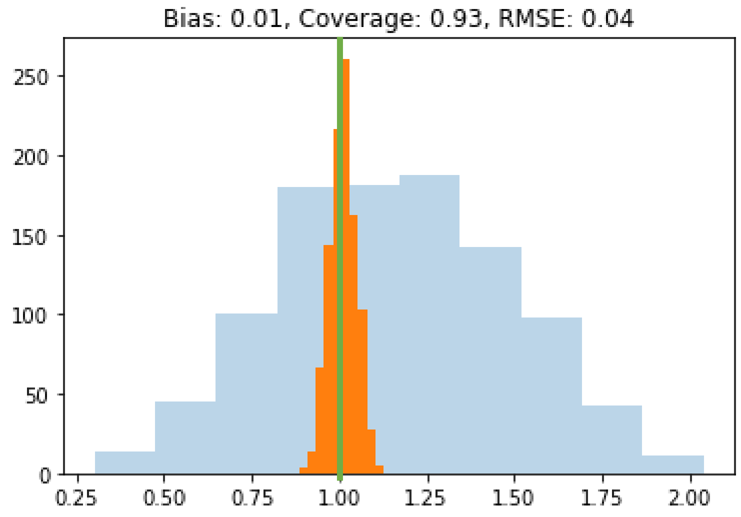
```
def dml2(X, D, y): # orthogonal dml with sample-splitting
    est_y = RandomForestRegressor(min_samples_leaf=20)
    yres = y - cross_val_predict(est_y, X, y, cv=3)
    est_t = RandomForestRegressor(min_samples_leaf=20)
    Dres = D - cross_val_predict(est_t, X, D, cv=3)
    theta = np.mean(yres * Dres) / np.mean(Dres**2)
    var = np.mean((Dres**2) * (yres - theta*Dres)**2) / np.mean(Dres**2)
    stderr = np.sqrt(var / X.shape[0])
    return theta, stderr
```



Natural Algorithm (Draft 3) Gone Right

```
from econml.dml import LinearDML

dml = LinearDML(model_y=RandomForestRegressor(min_samples_leaf=20),
                model_t=RandomForestRegressor(min_samples_leaf=20))
est.fit(y, D, W=X).effect_inference()
```



Stacking and Model Selection

- If we want to choose among many models or perform stacking, we can just use a stacked or automl model in place of each ML model

Stacking ML Models

```
def dml2(X, D, y): # orthogonal dml with sample-splitting
    est_y = StackingRegressor([rf, nnet, gbf, lasso])
    yres = y - cross_val_predict(est_y, X, y, cv=3)
    est_t = StackingRegressor([rf, nnet, gbf, lasso])
    Dres = D - cross_val_predict(est_t, X, D, cv=3)
    theta = np.mean(yres * Dres) / np.mean(Dres**2)
    var = np.mean((Dres**2) * (yres - theta*Dres)**2) / np.mean(Dres**2)
    stderr = np.sqrt(var / X.shape[0])
    return theta, stderr
```

AutoML Models

```
from flaml import AutoML

def dml2(X, D, y): # orthogonal dml with sample-splitting
    est_y = AutoML()
    yres = y - cross_val_predict(est_y, X, y, cv=3)
    est_t = AutoML()
    Dres = D - cross_val_predict(est_t, X, D, cv=3)
    theta = np.mean(yres * Dres) / np.mean(Dres**2)
    var = np.mean((Dres**2) * (yres - theta*Dres)**2) / np.mean(Dres**2)
    stderr = np.sqrt(var / X.shape[0])
    return theta, stderr
```

Stacking and Model Selection

- If we want to choose among many models or perform stacking, we can just use a stacked or automl model in place of each ML model
- Model selection or stacking done many times within each training fold
- Computationally expensive and statistically lossy

- Can we use all the data to at least select among models?

Semi-Cross-fitting Estimation Algorithm

- Split the data in half *(in practice K folds)*
- On first half, estimate $\hat{g}_1^{(1)}, \dots, \hat{g}_1^{(\mathrm{L})}$ of g_0 and predict on second half
- On second half, estimate $\hat{g}_2^{(1)}, \dots, \hat{g}_2^{(\mathrm{L})}$ of g_0 and predict on first half
- Choose the model $\ell \in \{1, \dots, L\}$ that optimizes out-of-sample RMSE
- On all data, solution $\hat{\theta}$ to empirical plug-in moment equation:

$$M_n(\hat{\theta}, \hat{g}) \coloneqq \frac{1}{n} \sum_{i \in S_1} m\left(Z_i; \hat{\theta}, \hat{g}_2^{(\ell)}\right) + \frac{1}{n} \sum_{i \in S_2} m\left(Z_i; \hat{\theta}, \hat{g}_1^{(\ell)}\right) = 0$$

Semi-Crossfitting

```
def dml2(X, D, y): # orthogonal dml with semi-crossfitting
    # cross val predict with many models
    est_y = [rf, gbf, lasso]
    yres = np.array([y - cross_val_predict(est, X, y, cv=3) for est in est_y])
    est_d = [rf, gbf, lasso]
    Dres = np.array([D - cross_val_predict(est, X, D, cv=3) for est in est_d])
    # select models with best out of fold performance
    best_y = np.argmin(np.mean(yres**2, axis=1))
    best_d = np.argmin(np.mean(Dres**2, axis=1))
    yres = yres[best_y]
    Dres = Dres[best_d]
    # go with their corresponding residuals
    theta = np.mean(yres * Dres) / np.mean(Dres**2)
    var = np.mean((Dres**2) * (yres - theta*Dres)**2) / np.mean(Dres**2)
    stderr = np.sqrt(var / X.shape[0])
    return theta, stderr
```

Semi-Crossfitting

- If the number of models L is small, then "spillover" is ok and approach still works. For practical purposes L should be thought as constant.
- Under further regularity, provably asymptotic normality holds if

$$\sqrt{\log(L)} = \mathrm{o}\left(\mathrm{n}^{1/4}\right)$$

Semi-Cross-fitting with Stacking

- Split the data in half *(in practice K folds)*
- On first half, estimate $\hat{g}_1^{(1)}, \dots, \hat{g}_1^{(\mathrm{L})}$ of g_0 and predict on second half
- On second half, estimate $\hat{g}_2^{(1)}, \dots, \hat{g}_2^{(\mathrm{L})}$ of g_0 and predict on first half
- Construct weights $\alpha_1, \dots, \alpha_\ell$ on the models using all the data (stacking)

- On all data, solution $\hat{\theta}$ to empirical plug-in moment equation:

$$M_n(\hat{\theta}, \hat{g}) := \frac{1}{n} \sum_{i \in S_1} m\left(Z_i; \hat{\theta}, \hat{g}_2^{(\ell)}\right) + \frac{1}{n} \sum_{i \in S_2} m\left(Z_i; \hat{\theta}, \hat{g}_1^{(\ell)}\right) = 0$$

Semi-Crossfitting with Stacking

```
def dml2(X, D, y): # orthogonal dml with semi-crossfitting and stacking
    # cross val predict with many models
    est_y = [rf, gbf, lasso]
    ypreds = np.array([cross_val_predict(est, X, y, cv=3) for est in est_y]).T
    est_d = [rf, gbf, lasso]
    Dpreds = np.array([cross_val_predict(est, X, D, cv=3) for est in est_d]).T
    # calculate stacked residuals by finding optimal coefficients
    # and weigthing out-of-sample predictions by these coefficients
    yres = y - LinearRegression().fit(ypreds, y).predict(ypreds)
    Dres = D - LinearRegression().fit(Dpreds, D).predict(Dpreds)
    # go with the stacked residuals
    theta = np.mean(yres * Dres) / np.mean(Dres**2)
    var = np.mean((Dres**2) * (yres - theta*Dres)**2) / np.mean(Dres**2)
    stderr = np.sqrt(var / X.shape[0])
    return theta, stderr
```

Semi-Crossfitting

- If the number of models L is small, then "spillover" is ok and approach still works. For practical purposes L should be thought as constant.
- Under further regularity, provably asymptotic normality holds if

$$\sqrt{L} = \mathrm{o}\left(\mathrm{n}^{1/4}\right)$$

Equivalent view of cross-fitting with stacking (lens of FWL theorem)

- Construct out of fold predictions based on many ML models
- Use these predictions as engineered features X in a simple OLS regression on D, X
- Use the coefficient and standard error of D from this final OLS

Proving the Main Theorem

Linear in θ Moments

- We will restrict attention to a broad class that simplifies proof

- Moment is linear in target parameter

$$m(Z;\theta,g) = a(Z;g)'\theta + \nu(Z;g)$$

- Expected moment also linear in θ

$$M(\theta,g) = A(g)'\theta + V(g)$$

Proof Ingredients: Linear in θ Moments

- Since $M_n(\hat{\theta}, \hat{g}) = 0$ we expect by concentration and sample splitting $M(\hat{\theta}, \hat{g}) \approx n^{-1/2}$
- Since $M(\theta_0, g_0) = 0$ we expect by Neyman orthogonality $M(\theta_0, \hat{g}) \approx RMSE(\hat{g})^2 = o(n^{-1/2})$
- Since moment is linear in θ: $A(\hat{g})\left(\hat{\theta} - \theta_0\right) = M(\hat{\theta}, \hat{g}) - M(\theta_0, \hat{g})$
- Since A is Lipschitz and $\hat{g} \to g_0$: $A(g_0)\left(\hat{\theta} - \theta_0\right) = M(\hat{\theta}, \hat{g}) - M(\theta_0, \hat{g}) + \mathrm{o_p}(\|\hat{\theta} - \theta_0\|)$
- Since $A(g_0)$ is invertible: $\|\hat{\theta} - \theta_0\| = \mathrm{O}(\|M(\hat{\theta}, \hat{g})\| + \|M(\theta_0, \hat{g})\|) = O_p(n^{-1/2})$

- More fine-grained analysis of $M(\hat{\theta}, \hat{g})$ term, shows: $\sqrt{n} M(\hat{\theta}, \hat{g}) \to \mathrm{N}(0, \mathrm{V})$

Proof of Main Theorem (visually)

CLT
+
sample-splitting
+
concentration
+
$\hat{g} \to g_0$

$M_n(\hat{\theta}, \hat{g}) = 0$

$N(0, V) \leftarrow$

$A(\hat{g})\left(\hat{\theta} - \theta_0\right) = M(\hat{\theta}, \hat{g}) - M(\theta_0, \hat{g})$

2nd order Taylor
+
orthogonality
+
$\|\hat{g} - g_0\| = o_p\left(n^{-1/4}\right)$

$= o_p\left(n^{-1/2}\right)$

$o_p(\|\hat{\theta} - \theta_0\|) =$

$A(g_0)\left(\hat{\theta} - \theta_0\right)$

$M(\theta_0, g_0) = 0$

Lipschitz $A(g)$
+
$\hat{g} \to g_0$

Proof of Main Theorem (algebraically)

- Since moment $M(\theta, g)$ is linear in θ and $M_n(\hat{\theta}, \hat{g}) = 0$ and $M(\theta_0, g_0) = 0$

$$\begin{aligned} A(\hat{g})\left(\hat{\theta} - \theta_0\right) &= M\left(\hat{\theta}, \hat{g}\right) - M(\theta_0, \hat{g}) \\ &= M\left(\hat{\theta}, \hat{g}\right) - M_n\left(\hat{\theta}, \hat{g}\right) + M(\theta_0, g_0) - M(\theta_0, \hat{g}) \end{aligned}$$

- Since $\mathrm{RMSE}(\hat{g}) = \|\hat{g} - g_0\| = o_p(1)$

$$A(g_0)\left(\hat{\theta} - \theta_0\right) = A(\hat{g})\left(\hat{\theta} - \theta_0\right) + \mathrm{o_p}\left(\|\hat{\theta} - \theta_0\|\right)$$

- Thus

$$A(g_0)\left(\hat{\theta} - \theta_0\right) = \underbrace{M\left(\hat{\theta}, \hat{g}\right) - M_n\left(\hat{\theta}, \hat{g}\right)}_{\substack{\to N(0, V) \\ \text{via CLT} \\ + \text{ sample-splitting} \\ + \text{ concentration} \\ + \hat{g} \to g_0}} + \underbrace{M(\theta_0, g_0) - M(\theta_0, \hat{g})}_{\substack{= o_p\left(n^{-1/2}\right) \\ \text{via orthogonality} \\ + \|\hat{g} - g_0\| = o_p\left(n^{-1/4}\right)}} + \mathrm{o_p}\left(\|\hat{\theta} - \theta_0\|\right)$$

Proof of Main Theorem: Orthogonality

- By Neyman orthogonality and bounded second derivative of $M(\theta_0, g)$ w.r.t. g
  $M(\theta_0, g_0) - M(\theta_0, \hat{g}) = D_g M(\theta_0, g_0)[g_0 - g] + O\left(\|\hat{g} - g_0\|^2\right) = o_p\left(n^{-1/2}\right)$

- Thus
$$A(g_0)\left(\hat{\theta} - \theta_0\right) = \underbrace{M\left(\hat{\theta}, \hat{g}\right) - M_n\left(\hat{\theta}, \hat{g}\right)}_{G_n\left(\hat{\theta}, \hat{g}\right)} + \mathrm{o_p}\left(n^{-1/2} + \left\|\hat{\theta} - \theta_0\right\|\right)$$

Proof of Main Theorem: Sample-Splitting (1)

- Let $G_n(\theta, g) = M(\theta, g) - M_n(\theta, g)$
$$G_n\left(\hat{\theta}, \hat{g}\right) = G_n(\theta_0, g_0) + G_n\left(\hat{\theta}, \hat{g}\right) - G_n(\theta_0, \hat{g}) + G_n(\theta_0, \hat{g}) - G_n(\theta_0, g_0)$$

- Linearity of moment + (sample-splitting and concentration $\Rightarrow$ $\|A(\hat{g}) - A_n(\hat{g})\| = o_p(1)$):
$$G_n\left(\hat{\theta}, \hat{g}\right) - G_n(\theta_0, \hat{g}) = \left(A(\hat{g}) - A_n(\hat{g})\right)\left(\hat{\theta} - \theta_0\right) = o_p\left(\left\|\hat{\theta} - \theta_0\right\|\right)$$

- Thus
$$A(g_0)\left(\hat{\theta} - \theta_0\right) = G_n(\theta_0, g_0) + G_n(\theta_0, \hat{g}) - G_n(\theta_0, g_0) + \mathrm{o_p}\left(n^{-1/2} + \left\|\hat{\theta} - \theta_0\right\|\right)$$

Proof of Main Theorem: Sample-Splitting (2)

- Note for $X_i = m(Z_i; \theta_0, \hat{g}) - m(Z_i; \theta_0, \hat{g}) - E[m(Z_i; \theta_0, \hat{g}) - m(Z_i; \theta_0, \hat{g})]$

$$G_n(\theta_0, \hat{g}) - G_n(\theta_0, g_0) = \frac{1}{n_2} \sum_{i \in S_2} X_i$$

- By sample splitting, X_i are i.i.d. with $E[X_i] = 0$. By variance decomposition (concentration)

$$\left\| \frac{1}{n_2} \sum_{i \in S_2} X_i \right\|_{L_2} \leq \sqrt{\frac{E\left[X_i^2\right]}{n}}$$

- Thus

$$\|G_n(\theta_0, \hat{g}) - G_n(\theta_0, g_0)\|_{L_2} \leq \frac{1}{\sqrt{n}} \sqrt{E\left[\left(m(Z_i; \theta_0, \hat{g}) - m(Z_i; \theta_0, \hat{g})\right)^2\right]} = O\left(\frac{\|\hat{g} - g_0\|}{\sqrt{n}}\right) = o_p\left(n^{-1/2}\right)$$

Concluding

- So far

$$A(g_0)\left(\hat{\theta}-\theta_0\right)=G_n(\theta_0,g_0)+\mathrm{o_p}\left(n^{-1/2}+\left\|\hat{\theta}-\theta_0\right\|\right)$$

- Since $A(g_0)$ is invertible and $G_n(\theta_0,g_0)=O_p\left(n^{-1/2}\right)$ by concentration

$$\left\|\hat{\theta}-\theta_0\right\|=O_p\left(n^{-1/2}\right)$$

- Thus, we have asymptotic linearity

$$\sqrt{n}\left(\hat{\theta}-\theta_0\right)=\sqrt{n}A(g_0)^{-1}G_n(\theta_0,g_0)+\mathrm{o_p}(1)=\frac{1}{\sqrt{n_2}}\sum_{i\in S_2}A(g_0)^{-1}m(Z_i;\theta_0,g_0)+o_p(1)$$

- By CLT we get the theorem

Main Theorem

- If moment is Neyman orthogonal and RMSE of $\hat{g}$ is $o_p\left(n^{-1/4}\right)*$

$$\sqrt{n}\,\left(\hat{\theta} - \theta_0\right) \to N\left(0, A^{-1}\Sigma\left(\mathrm{A}^{-1}\right)^{\top}\right)$$

- $A = \nabla_\theta M(\theta_0, g_0)$ and $\Sigma = E\left[m(Z;\theta_0, g_0)\, m(Z;\theta_0, g_0)^{\top}\right]$

*plus regularity conditions