ML in Economics and Finance: Where do We Go Now? - Part II

Raul Riva

FGV EPGE

December, 2025

INSPER - São Paulo

Flight Plan ✈️

1. What is ML, anyway?
2. Causality in High Dimensions

$\Big\}$ **Yesterday**

3. (Seriously) Heterogeneous Partial Effects
4. Solving Large-Scale General Equilibrium Models

$\Big\}$ **Today**

Heterogeneous Partial Effects

Motivation

Let Y be an outcome and X, Z be features (covariates). We frequently want to approximate

$$h(x, z) \equiv \mathbb{E}[Y|X = x, Z = z]$$

and the **partial effects**

$$\frac{\partial}{\partial x} h(x, z) = \frac{\partial}{\partial x} \mathbb{E}[Y|X = x, Z = z].$$

Motivation

Let Y be an outcome and X, Z be features (covariates). We frequently want to approximate

$$h(x, z) \equiv \mathbb{E}[Y|X = x, Z = z]$$

and the **partial effects**

$$\frac{\partial}{\partial x} h(x, z) = \frac{\partial}{\partial x} \mathbb{E}[Y|X = x, Z = z].$$

- This is a prediction problem after all!

Motivation

Let Y be an outcome and X, Z be features (covariates). We frequently want to approximate

$$h(x, z) \equiv \mathbb{E}[Y|X = x, Z = z]$$

and the **partial effects**

$$\frac{\partial}{\partial x} h(x, z) = \frac{\partial}{\partial x} \mathbb{E}[Y|X = x, Z = z].$$

- This is a prediction problem after all!
- Approach 1: impose a parametric model for h, e.g. linear regression. Pros and cons?
- Approach 2: use fully nonparametric methods. Pros and cons?

Motivation

Let Y be an outcome and X, Z be features (covariates). We frequently want to approximate

$$h(x,z) \equiv \mathbb{E}[Y|X=x, Z=z]$$

and the **partial effects**

$$\frac{\partial}{\partial x} h(x,z) = \frac{\partial}{\partial x}\mathbb{E}[Y|X=x, Z=z].$$

- This is a prediction problem after all!
- Approach 1: impose a parametric model for h, e.g. linear regression. Pros and cons?
- Approach 2: use fully nonparametric methods. Pros and cons?
- The third way is the charm: a bit of structure, a bit of ML!

Example I - Heterogenous Treatment Effects

- Outcomes Y_i depend on a treatment $X_i \in \mathbb{R}$ and covariates $Z_i \in \mathbb{R}^p$;
- The dose X_i depends on observables Z_i;
- Potential outcomes $Y_i(x)$ for each dose $x \in \mathbb{R}$;

Example I - Heterogenous Treatment Effects

- Outcomes Y_i depend on a treatment $X_i \in \mathbb{R}$ and covariates $Z_i \in \mathbb{R}^p$;
- The dose X_i depends on observables Z_i;
- Potential outcomes $Y_i(x)$ for each dose $x \in \mathbb{R}$;
- The conditional average effect of increasing the dose is $\tau(x, z) \equiv \frac{\partial}{\partial x}\mathbb{E}[Y(x)|Z = z]$;
- If $\mathbb{E}[Y(x)|X = x, Z = z] = \mathbb{E}[Y(x)|Z = z]$ (common assumption in the literature), then

$$\tau(x, z) = \frac{\partial}{\partial x}\mathbb{E}[Y|X = x, Z = z] = \frac{\partial h(x, z)}{\partial x}$$

Example II - Grouped Heterogeneity

Consider the following model:

$$Y_i = \alpha(Z_i) + X_i'\beta + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i | X_i, Z_i] = 0,$$

- X_i affects Y_i homogeneously;
- Intercept $\alpha(Z_i)$ varies with Z_i, maybe in a highly nonlinear way;
- Since Z_i and X_i can be correlated, this can affect inference about β;
- Bonhomme and Manresa (2015) studied how democracy affects national income using this model;
- In their case: $\alpha(Z_i)$ is constant across groups but Z_i defines membership;
- In our notation: $h(z, x) = \alpha(z) + x'\beta$

How can we balance flexibility and interpretability?

Masini and Medeiros (2025) 🇧🇷 proposed a middle ground:

$$h(x,z) = x^\top \beta(z), \qquad \frac{\partial h(x,z)}{\partial x} = \beta(z)$$

- The partial effect of X on Y varies with Z through $\beta(z)$;
- $\beta(.)$ is a Lipschitz function that can be estimated with ML methods;
- No need to numerically approximate $\frac{\partial}{\partial x} h(x,z)$;

How can we balance flexibility and interpretability?

Masini and Medeiros (2025) 🇧🇷 proposed a middle ground:

$$h(x, z) = x^\top \beta(z), \qquad \frac{\partial h(x, z)}{\partial x} = \beta(z)$$

- The partial effect of X on Y varies with Z through $\beta(z)$;
- $\beta(.)$ is a Lipschitz function that can be estimated with ML methods;
- No need to numerically approximate $\frac{\partial}{\partial x} h(x, z)$;
- Explicit conditions for consistency *and* asymptotic normality of $\hat{\beta}(z)$;

How can we balance flexibility and interpretability?

Masini and Medeiros (2025) 🇧🇷 proposed a middle ground:

$$h(x, z) = x^\top \beta(z), \qquad \frac{\partial h(x, z)}{\partial x} = \beta(z)$$

- The partial effect of X on Y varies with Z through $\beta(z)$;
- $\beta(.)$ is a Lipschitz function that can be estimated with ML methods;
- No need to numerically approximate $\frac{\partial}{\partial x} h(x, z)$;
- Explicit conditions for consistency *and* asymptotic normality of $\hat{\beta}(z)$;
- Secrete sauce: a variant of the **Random Forest** algorithm!

How can we balance flexibility and interpretability?

Masini and Medeiros (2025) 🇧🇷 proposed a middle ground:

$$h(x, z) = x^\top \beta(z), \qquad \frac{\partial h(x, z)}{\partial x} = \beta(z)$$

- The partial effect of X on Y varies with Z through $\beta(z)$;
- $\beta(.)$ is a Lipschitz function that can be estimated with ML methods;
- No need to numerically approximate $\frac{\partial}{\partial x} h(x, z)$;
- Explicit conditions for consistency *and* asymptotic normality of $\hat{\beta}(z)$;
- Secrete sauce: a variant of the **Random Forest** algorithm!

But what is a Random Forest, anyway? 🤔

Questions?

Quick Intro to Random Forests

Random Trees

Recall the general ML framework:

$$Y = f(X) + \varepsilon$$

A **Random Tree** is a particular way of parametrizing f!

Random Trees

Recall the general ML framework:

$$Y = f(X) + \varepsilon$$

A **Random Tree** is a particular way of parametrizing f!

- If $X \in \mathbb{R}^p$, consider a finite partition $\{S_1, S_2, \ldots, S_m\}$ of $\mathbb{R}^p$;
- Each S_i is a hyperrectangle defined by recursive binary splits on the covariates;
- On each S_i, f is constant: $f(x) = \mu_i$ for all $x \in S_i$;
- After a tree has been estimated ("grown"), $\hat{f}(x_i) = \mu_i$ if $x_i \in S_i$;

Random Trees

Recall the general ML framework:

$$Y = f(X) + \varepsilon$$

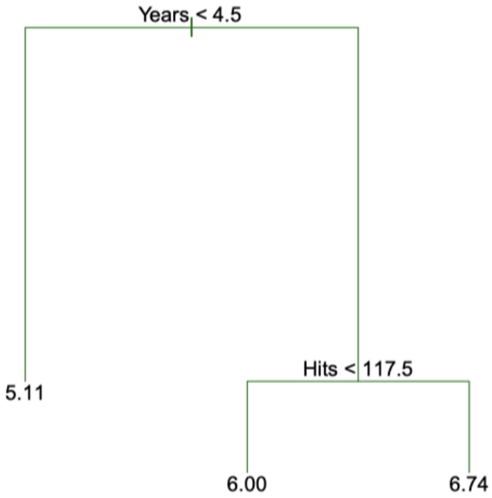
A **Random Tree** is a particular way of parametrizing f!

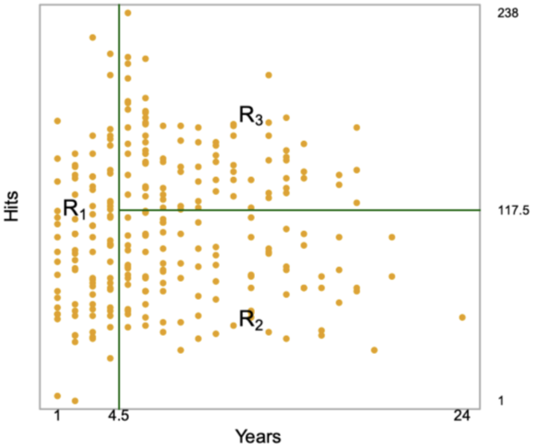
- If $X \in \mathbb{R}^p$, consider a finite partition $\{S_1, S_2, \dots, S_m\}$ of $\mathbb{R}^p$;
- Each S_i is a hyperrectangle defined by recursive binary splits on the covariates;
- On each S_i, f is constant: $f(x) = \mu_i$ for all $x \in S_i$;
- After a tree has been estimated ("grown"), $\hat{f}(x_i) = \mu_i$ if $x_i \in S_i$;

The really complicated part: there are *so many* partitions... how to pick one?

Hyafil and Rivest (1976): this is harder than you think! The problem is NP-complete!

Example: Predicting Baseball Player Salaries





How to pick a split point? Use some greed!

Let's say you want to split on feature X_j at point δ:

$$S_1 \equiv \{x \in \mathbb{R}^p : x_j \leq \delta\}, \quad S_2 \equiv \{x \in \mathbb{R}^p : x_j > \delta\}$$
$$\mu_1 \equiv \sum_{i: x_i \in S_1} \frac{Y_i}{n_1}, \quad \mu_2 \equiv \sum_{i: x_i \in S_2} \frac{Y_i}{n_2}$$

How to pick a split point? Use some greed!

Let's say you want to split on feature X_j at point δ:

$$S_1 \equiv \{x \in \mathbb{R}^p : x_j \leq \delta\}, \quad S_2 \equiv \{x \in \mathbb{R}^p : x_j > \delta\}$$
$$\mu_1 \equiv \sum_{i:x_i \in S_1} \frac{Y_i}{n_1}, \quad \mu_2 \equiv \sum_{i:x_i \in S_2} \frac{Y_i}{n_2}$$

- Define $SSR(\delta) \equiv \sum_{i:x_i \in S_1} (Y_i - \mu_1)^2 + \sum_{i:x_i \in S_2} (Y_i - \mu_2)^2$
- Choose δ to minimize $SSR(\delta) \implies$ this is usually very fast to compute!

How to pick a split point? Use some greed!

Let's say you want to split on feature X_j at point δ:

$$S_1 \equiv \{x \in \mathbb{R}^p : x_j \leq \delta\}, \quad S_2 \equiv \{x \in \mathbb{R}^p : x_j > \delta\}$$
$$\mu_1 \equiv \sum_{i:x_i \in S_1} \frac{Y_i}{n_1}, \quad \mu_2 \equiv \sum_{i:x_i \in S_2} \frac{Y_i}{n_2}$$

- Define $SSR(\delta) \equiv \sum_{i:x_i \in S_1} (Y_i - \mu_1)^2 + \sum_{i:x_i \in S_2} (Y_i - \mu_2)^2$
- Choose δ to minimize $SSR(\delta) \implies$ this is usually very fast to compute!
- Repeat this for all features X_j and pick the best one;
- Important: you need some stopping rule! There is a huge literature on this...
- Example: minimum number of observations per leaf;

How to pick a split point? Use some greed!

Let's say you want to split on feature X_j at point δ:

$$S_1 \equiv \{x \in \mathbb{R}^p : x_j \leq \delta\}, \quad S_2 \equiv \{x \in \mathbb{R}^p : x_j > \delta\}$$
$$\mu_1 \equiv \sum_{i:x_i \in S_1} \frac{Y_i}{n_1}, \quad \mu_2 \equiv \sum_{i:x_i \in S_2} \frac{Y_i}{n_2}$$

- Define $SSR(\delta) \equiv \sum_{i:x_i \in S_1} (Y_i - \mu_1)^2 + \sum_{i:x_i \in S_2} (Y_i - \mu_2)^2$
- Choose δ to minimize $SSR(\delta) \implies$ this is usually very fast to compute!
- Repeat this for all features X_j and pick the best one;
- Important: you need some stopping rule! There is a huge literature on this...
- Example: minimum number of observations per leaf;

This is the so-called the **CART** algorithm due to Breiman et al. (1984).

How to get a Random Forest?

A Random Forest is an **ensemble** of Random Trees:

$$\hat{f}^{(1)}(x), \hat{f}^{(2)}(x), \ldots, \hat{f}^{(B)}(x)$$

How to get a Random Forest?

A Random Forest is an **ensemble** of Random Trees:

$$\hat{f}^{(1)}(x), \hat{f}^{(2)}(x), \ldots, \hat{f}^{(B)}(x)$$

Each tree is grown on a **perturbed version** of the data:

- Bootstrap sample of the observations;
- Random subset of features considered at each split;

How to get a Random Forest?

A Random Forest is an **ensemble** of Random Trees:

$$\hat{f}^{(1)}(x), \hat{f}^{(2)}(x), \ldots, \hat{f}^{(B)}(x)$$

Each tree is grown on a **perturbed version** of the data:

- Bootstrap sample of the observations;
- Random subset of features considered at each split;

The forest prediction is the **average**:

$$\hat{f}_{RF}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{(b)}(x)$$

How to get a Random Forest?

A Random Forest is an **ensemble** of Random Trees:

$$\hat{f}^{(1)}(x), \hat{f}^{(2)}(x), \ldots, \hat{f}^{(B)}(x)$$

Each tree is grown on a **perturbed version** of the data:

- Bootstrap sample of the observations;
- Random subset of features considered at each split;

The forest prediction is the **average**:

$$\hat{f}_{RF}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{(b)}(x)$$

Key insight:

- Each tree is noisy and biased, but averaging them reduces variance dramatically;
- Randomness *decorrelates* the trees, making averaging powerful;

Questions?

Back to Partial Effects

The Main Insight

We have a random sample $\{(Y_i, X_i, Z_i)\}_{i=1}^n$ from

$$Y_i = X_i^\top \beta(Z_i) + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i | X_i, Z_i] = 0$$

The Main Insight

We have a random sample $\{(Y_i, X_i, Z_i)\}_{i=1}^{n}$ from

$$Y_i = X_i^\top \beta(Z_i) + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i | X_i, Z_i] = 0$$

- Masini and Medeiros (2025) proposed to estimate $\beta(z)$ using a modified Random Forest;
- Key modification: at each split, try to minimize the **local least squares** criterion;

The Main Insight

We have a random sample $\{(Y_i, X_i, Z_i)\}_{i=1}^n$ from

$$Y_i = X_i^\top \beta(Z_i) + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i | X_i, Z_i] = 0$$

- Masini and Medeiros (2025) proposed to estimate $\beta(z)$ using a modified Random Forest;
- Key modification: at each split, try to minimize the **local least squares** criterion;

Suppose you want to split at $Z_j \leq \delta$ as before. Then:

$$S_1 \equiv \{z \in \mathbb{R}^p : z_j \leq \delta\}, \quad S_2 \equiv \{z \in \mathbb{R}^p : z_j > \delta\}$$

$$(\hat{\beta}_1) \equiv \arg\min_{\beta} \sum_{i:Z_{i,j} \in S_1} (Y_i - X_i^\top \beta)^2, \quad (\hat{\beta}_2) \equiv \arg\min_{\beta} \sum_{i:Z_{i,j} \in S_2} (Y_i - X_i^\top \beta)^2$$

$$SSR(\delta) \equiv \sum_{i:Z_{i,j} \in S_1} (Y_i - X_i^\top \hat{\beta}_1)^2 + \sum_{i:Z_{i,j} \in S_2} (Y_i - X_i^\top \hat{\beta}_2)^2$$

The Main Insight

We have a random sample $\{(Y_i, X_i, Z_i)\}_{i=1}^n$ from

$$Y_i = X_i^\top \beta(Z_i) + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i | X_i, Z_i] = 0$$

- Masini and Medeiros (2025) proposed to estimate $\beta(z)$ using a modified Random Forest;
- Key modification: at each split, try to minimize the **local least squares** criterion;

Suppose you want to split at $Z_j \leq \delta$ as before. Then:

$$S_1 \equiv \{z \in \mathbb{R}^p : z_j \leq \delta\}, \quad S_2 \equiv \{z \in \mathbb{R}^p : z_j > \delta\}$$

$$(\hat{\beta}_1) \equiv \arg\min_{\beta} \sum_{i: Z_{i,j} \in S_1} (Y_i - X_i^\top \beta)^2, \quad (\hat{\beta}_2) \equiv \arg\min_{\beta} \sum_{i: Z_{i,j} \in S_2} (Y_i - X_i^\top \beta)^2$$

$$SSR(\delta) \equiv \sum_{i: Z_{i,j} \in S_1} (Y_i - X_i^\top \hat{\beta}_1)^2 + \sum_{i: Z_{i,j} \in S_2} (Y_i - X_i^\top \hat{\beta}_2)^2$$

Pick δ to minimize $SSR(\delta)$!

The Algorithm

Pick a number of trees B and a minimum leaf size k. For $b = 1, \dots, B$:

1. Draw a bootstrap sample of size $s \leq n$ from the data;
2. Divide the data into two halves $\mathcal{A}$ and $\mathcal{B}$;
3. Using $\mathcal{B}$, keep splitting at random dimensions j using the previous criterion;
4. Stop when all leaves have less than $2k - 1$ and more than k observations;
5. Using $\mathcal{A}$, estimate $\beta(z)$ using only observations in the leaf where z falls;

The final estimate is

$$\hat{\beta}(z) = \frac{1}{B} \sum_{b=1}^{B} \hat{\beta}^{(b)}(z)$$

The Algorithm

Pick a number of trees B and a minimum leaf size k. For $b = 1, \dots, B$:

1. Draw a bootstrap sample of size $s \leq n$ from the data;
2. Divide the data into two halves $\mathcal{A}$ and $\mathcal{B}$;
3. Using $\mathcal{B}$, keep splitting at random dimensions j using the previous criterion;
4. Stop when all leaves have less than $2k - 1$ and more than k observations;
5. Using $\mathcal{A}$, estimate $\beta(z)$ using only observations in the leaf where z falls;

The final estimate is

$$\hat{\beta}(z) = \frac{1}{B} \sum_{b=1}^{B} \hat{\beta}^{(b)}(z)$$

This algorithm uses **honest trees**! Similar intuition to cross-fitting.

Cool Properties and Limitations

Cool properties:

- Highly interpretable and relatively mild assumptions on $\beta(z)$;
- Easy confidence intervals for $\beta(z)$ at any point z:

$$\Omega^{-1/2}(z)\left(\hat{\beta}(z) - \beta(z)\right) \xrightarrow{d} \mathcal{N}(0, I_q)$$

  for some complicated $\Omega(x)$ that can be estimated consistently;
- There is also a Lagrange multiplier test for homogeneity of $\beta(z)$;

Cool Properties and Limitations

Cool properties:

- Highly interpretable and relatively mild assumptions on $\beta(z)$;
- Easy confidence intervals for $\beta(z)$ at any point z:

$$\Omega^{-1/2}(z)\left(\hat{\beta}(z)-\beta(z)\right) \xrightarrow{d} \mathcal{N}(0, I_q)$$

  for some complicated $\Omega(x)$ that can be estimated consistently;
- There is also a Lagrange multiplier test for homogeneity of $\beta(z)$;

Limitations:

- The dimension of X_i should be small relative to n;
- The dimension of Z_i cannot be *that* large relative to n;
- Pointwise inference only;
- It cannot be readily applied to time series and panel data;
- It can be computationally demanding in large datasets;

Questions?

Solving Large-Scale General Equilibrium Models

Thank you!

See you tomorrow, stay tuned!

Appendix and References

References

Bonhomme, Stéphane and Elena Manresa (May 2015). "Grouped Patterns of Heterogeneity in Panel Data: Grouped Patterns of Heterogeneity". In: *Econometrica* 83.3, pp. 1147–1184. ISSN: 0012-9682. DOI: 10.3982/ecta11319. URL: http://dx.doi.org/10.3982/ECTA11319.

Breiman, Leo, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone (1984). *Classification and Regression Trees*. en.

Hyafil, Laurent and Ronald L. Rivest (May 1976). "Constructing optimal binary decision trees is NP-complete". In: *Information Processing Letters* 5.1, pp. 15–17. ISSN: 0020-0190. DOI: 10.1016/0020-0190(76)90095-8. URL: http://dx.doi.org/10.1016/0020-0190(76)90095-8.

Masini, Ricardo and Marcelo Medeiros (2025). *Balancing Flexibility and Interpretability: A Conditional Linear Model Estimation via Random Forest*. DOI: 10.48550/ARXIV.2502.13438. URL: https://arxiv.org/abs/2502.13438.