What if the cure is worse than the disease?

Policy learning for maximizing a primary outcome while controlling an adverse event

Laura Fuentes, M. Even, J. Josse, A. Chambaz

56es Journées de Statistique de la SFdS, Marseille 2025

05/06/2025

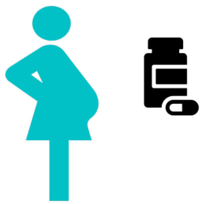
I. Context

I.1-Medical motivations

Classical policy learning: Given **patient's characteristics**,
determine the **optimal treatment maximizing** each patient's outcome

IVF example:

Find the **optimal hormone dose** to **maximize** the number of oocyte produced



I.1-Medical motivations

Classical policy learning: Given **patient's characteristics**,
determine the **optimal treatment maximizing** each patient's outcome

> **IVF example:**
>
> Find the **optimal hormone dose** to **maximize** the number of oocyte produced

Our goal: Given **patient's characteristics**,
determine the **optimal treatment maximizing** each patient's outcome
while controlling an **adverse event**

> **IVF example:**
>
> Find the **optimal hormone dose** to **maximize** the number of oocyte produced
> while avoiding **ovarian hyperstimulation**

I.2-Mathematical framework

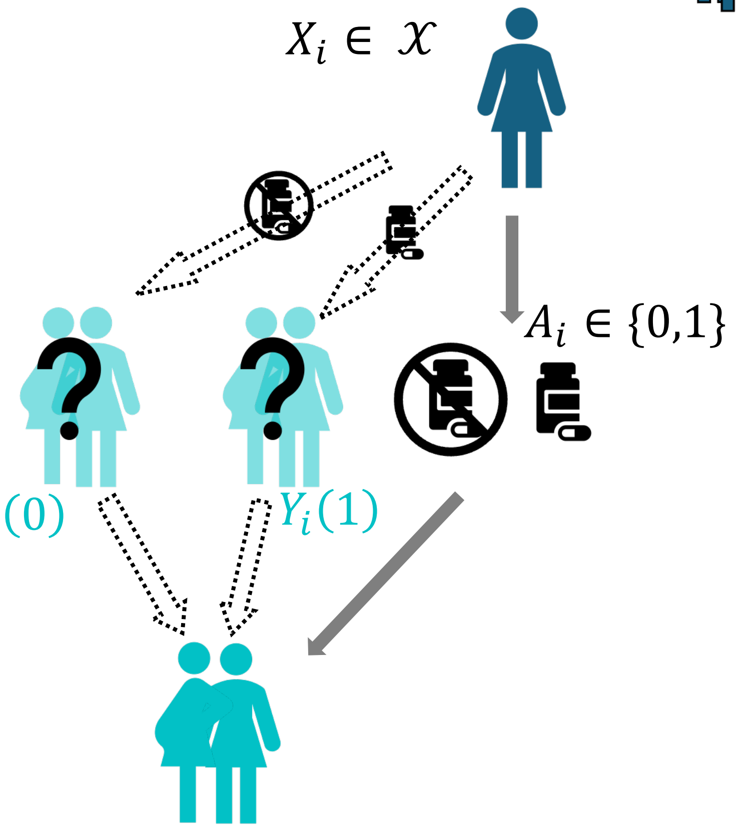
Set of independent and identically distributed subjects

Covariates:
$X_i \in \mathcal{X}$

Binary treatment:
$A_i \in \{0,1\}$

I.2-Mathematical framework

Set of independent and identically distributed subjects

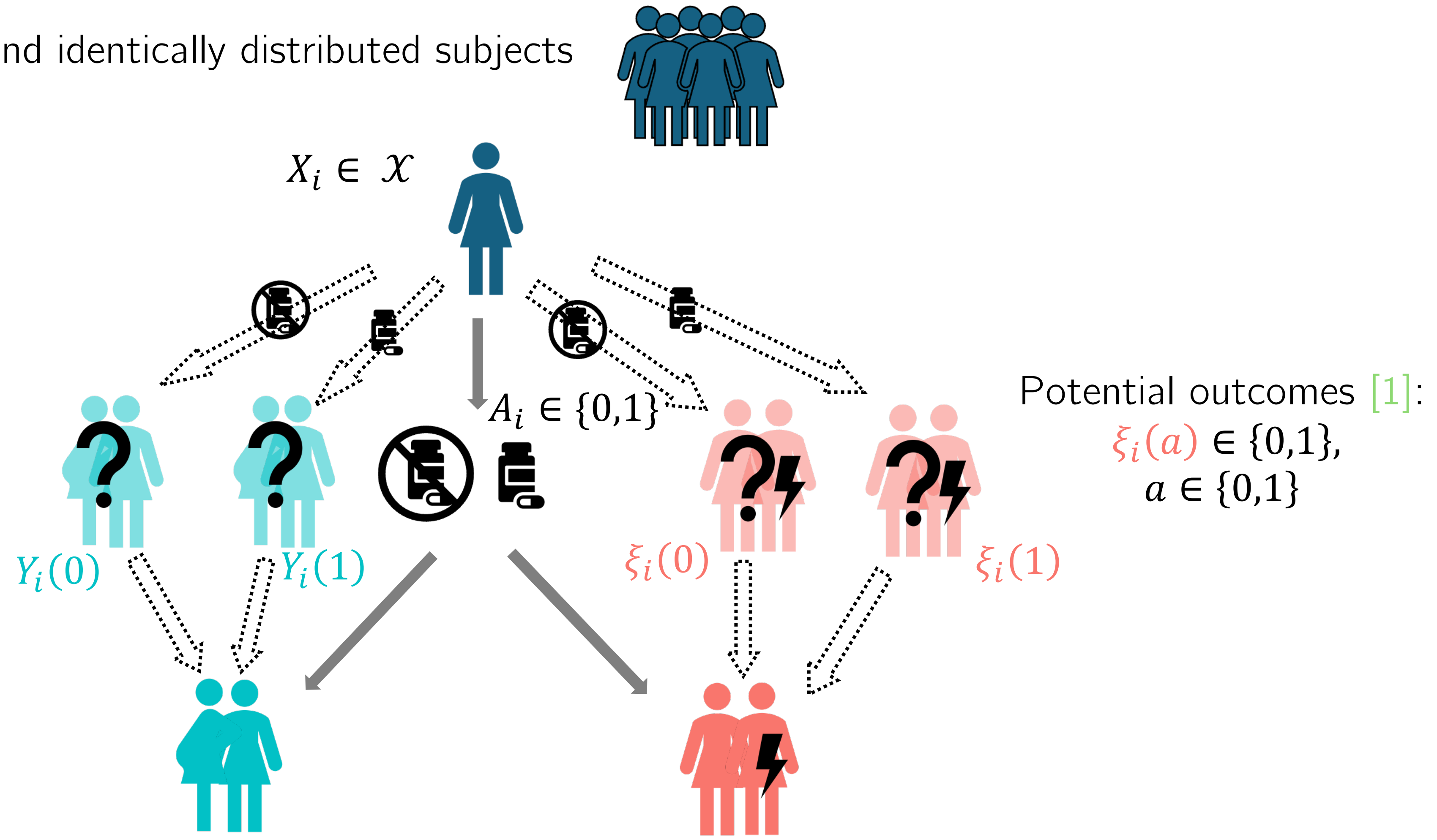


Potential outcomes [1]:
$Y_i(a) \in [0,1],\ a \in \{0,1\}$

Observed primary outcome:
$Y_i = Y_i(A_i) \in [0,1]$

I.2-Mathematical framework

Set of independent and identically distributed subjects

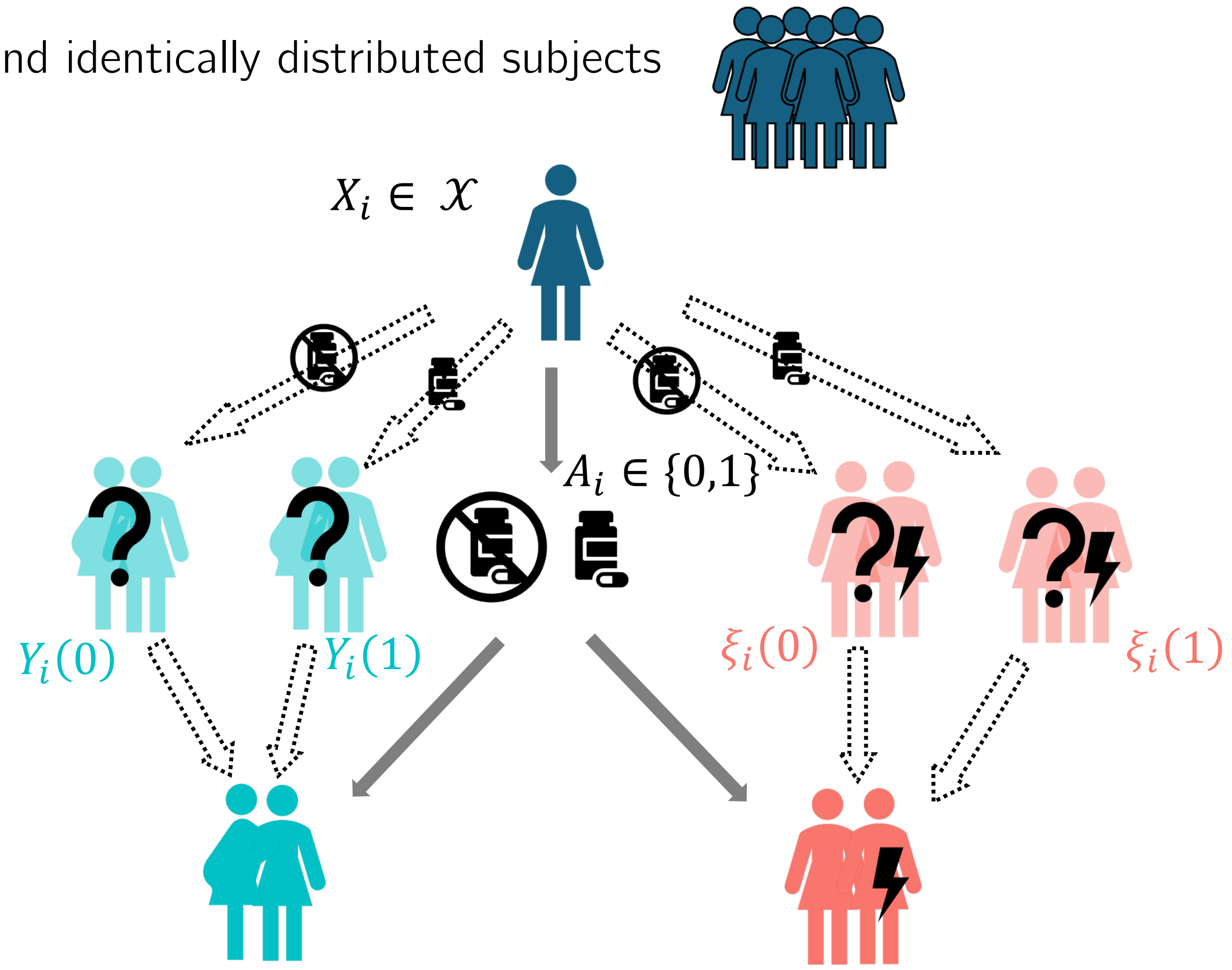


Observed primary outcome:
$Y_i = Y_i(A_i) \in [0,1]$

Observed adverse event:
$\xi_i = \xi_i(A_i) \in \{0,1\}$

I.2-Mathematical framework

Set of independent and identically distributed subjects



Observed primary outcome: $Y_i = Y_i(A_i) \in [0,1]$

Observed adverse event: $\xi_i = \xi_i(A_i) \in \{0,1\}$

$$\mathcal{O}_i = (X_i, A_i, Y_i, \xi_i) \sim P$$

I.2-Policy learning framework

Policy: $\pi \in \Pi$ maps patient's characteristics to a treatment decision

$\pi : X \to \{0,1\}$ or $[0,1]$

I.2-Policy learning framework

Policy: $\pi \in \Pi$ maps patient's characteristics to a treatment decision

$\pi : X \to \{0,1\}$ or $[0,1]$

Classical policy learning: Find the value-optimal policy π^*, maximizing the average outcome if π is followed,

$$x \mapsto \pi^*(x) = \mathbf{1}\{\Delta\mu(x) > 0\}$$

$$\mu(a, x) = E[Y|a, x]$$
$$\Delta\mu(x) = \mu(1, x) - \mu(0, x)$$

Covariates	Treatment	$\mu(0, X)$	$\mu(1, X)$	$\Delta\mu(X)$	$\pi^*(X)$
4 | 1 | a	1	**NA**	0.9	**NA**	?
10 | 0 | b	0	0.6	**NA**	**NA**	?
2 | 1 | a	0	0.3	**NA**	**NA**	?

I.2-Policy learning framework

Policy: $\pi \in \Pi$ maps patient's characteristics to a treatment decision $\quad \pi : X \to \{0,1\}$ or $[0,1]$

<u>Classical policy learning</u>: Find the value-optimal policy π^*, maximizing the average outcome if π is followed,

$$x \mapsto \pi^*(x) = \mathbf{1}\{\Delta\mu(x) > 0\}$$

$\mu(a,x) = E[Y|a,x]$
$\Delta\mu(x) = \mu(1,x) - \mu(0,x)$

1- <u>Direct</u>: Predict sign of $\Delta\mu(X)$

2- <u>Indirect</u>: Plug-in an estimation of $\Delta\mu$

$$x \mapsto \hat{\pi}^*(x) = \mathbf{1}\{\hat{\Delta}\mu_n(x) > 0\}$$

Covariates	Treatment	$\hat{\mu}_n(0,X)$	$\hat{\mu}_n(1,X)$	$\hat{\Delta}\mu_n(X)$	$\hat{\pi}^*(X)$
4 | 1 | a	1	0.7	0.9	0.2	1
10 | 0 | b	0	0.6	0.3	-0.3	0
2 | 1 | a	0	0.3	0.4	0.1	1

I.2-Policy learning framework

Policy: $\pi \in \Pi$ maps patient's characteristics to a treatment decision $\pi : X \to \{0,1\}$ or $[0,1]$

Classical policy learning: Find the value-optimal policy π^*, maximizing the average outcome if π is followed,

$$x \mapsto \pi^*(x) = \mathbf{1}\{\Delta\mu(x) > 0\}$$

$$\mu(a,x) = E[Y|a,x]$$
$$\Delta\mu(x) = \mu(1,x) - \mu(0,x)$$

1- Direct: Predict sign of $\Delta\mu(X)$

2- Indirect: Plug-in an estimation of $\Delta\mu$

$$x \mapsto \hat{\pi}^*(x) = \mathbf{1}\{\hat{\Delta}\mu_n(x) > 0\}$$

Covariates	Treatment	$\hat{\mu}_n(\mathbf{0},\boldsymbol{X})$	$\hat{\mu}_n(\mathbf{1},\boldsymbol{X})$	$\hat{\Delta}\mu_n(\boldsymbol{X})$	$\hat{\pi}^*(\boldsymbol{X})$
4 | 1 | a	1	0.7	0.9	0.2	1
10 | 0 | b	0	0.6	0.3	-0.3	0
2 | 1 | a	0	0.3	0.4	0.1	1

❌ Relies on quality of estimator $\hat{\mu}_n$

Problematic values with some estimators

II. Methods

II.1- Problem formulation

EP-learning [2]: Efficient plug-in risk estimator

$$R: \psi \mapsto E[\psi(X)^2 - 2\psi(X) \cdot \Delta\mu(X)] \quad (1)$$

1- Plug-in efficient estimator of $\Delta\mu$

2- Minimize risk (1) for $\psi \in \Psi$ (convex space)

❌ Overlooks adverse events

II.1- Problem formulation

EP-learning [2]: Efficient plug-in risk estimator

1- Plug-in efficient estimator of $\Delta\mu$

2- Minimize risk (1) for $\psi \in \Psi$

$$R: \psi \mapsto E[\psi(X)^2 - 2\psi(X) \cdot \Delta\mu(X)] \quad (1)$$

❌ Overlooks adverse events

Add constraint: Guarantee that π does **not increase** the average probability of adverse event beyond $\alpha \in \left[0, \frac{1}{2}\right]$

$$\left| \begin{array}{l} \nu(A, X) = E[\xi | A, X] \\ \Delta\nu(X) = \nu(1, X) - \nu(0, X) \end{array} \right.$$

II.1- Problem formulation

EP-learning [2]: Efficient plug-in risk estimator

1- Plug-in efficient estimator of $\Delta\mu$

2- Minimize risk (1) for $\psi \in \Psi$

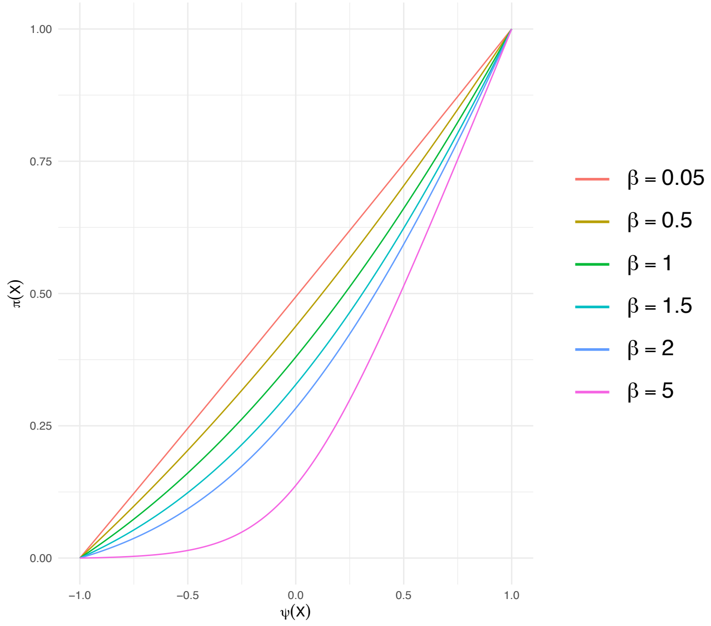
$$R: \psi \mapsto E[\psi(X)^2 - 2\psi(X) \cdot \Delta\mu(X)] \quad (1)$$

Add constraint: Guarantee that π does **not increase** the average probability of adverse event beyond $\alpha \in \left[0, \frac{1}{2}\right]$

$$\nu(A, X) = E[\xi | A, X]$$
$$\Delta\nu(X) = \nu(1, X) - \nu(0, X)$$

Policy reformulation:

$$\pi(X) = \sigma_\beta \circ \psi(X) \in [0,1]$$



✓ Overcome differentiability issues & strongly convex

✓ Confidence measure in treatment recommendation

II.1- Problem formulation

EP-learning [2]: Efficient plug-in risk estimator

1- Plug-in efficient estimator of $\Delta\mu$

2- Minimize risk (1) for $\psi \in \Psi$

$$R: \psi \mapsto E[\psi(X)^2 - 2\psi(X) \cdot \Delta\mu(X)] \quad (1)$$

Add constraint: Guarantee that π does **not increase** the average probability of adverse event beyond $\alpha \in \left[0, \frac{1}{2}\right]$

$$S: \psi \mapsto E\left[\underbrace{\sigma_\beta \circ \psi(X)}_{\pi(X)} \cdot \Delta\nu(X)\right] - \alpha \quad (2)$$

$\nu(A, X) = E[\xi | A, X]$
$\Delta\nu(X) = \nu(1, X) - \nu(0, X)$

Assumption: Treated patients are more likely to suffer from an adverse event, that is: $\Delta\nu(X) \geq 0$

II.1- Problem formulation

EP-learning [2]: Efficient plug-in risk estimator

1- Plug-in efficient estimator of $\Delta\mu$

2- Minimize risk (1) for $\psi \in \Psi$

$$R: \psi \mapsto E[\psi(X)^2 - 2\psi(X) \cdot \Delta\mu(X)] \quad (1)$$

Add constraint: Guarantee that π does **not increase** the average probability of adverse event beyond $\alpha \in \left[0, \frac{1}{2}\right]$

$$S: \psi \mapsto E\left[\sigma_\beta \circ \psi(X) \cdot \Delta\nu(X)\right] - \alpha \quad (2)$$

$$\nu(A, X) = E[\xi | A, X]$$
$$\Delta\nu(X) = \nu(1, X) - \nu(0, X)$$

Policy optimization objective

Minimize $R(\psi)$ w.r.t $\psi \in \Psi$
s.t. $S(\psi) \leq 0$

$$\xrightarrow{\text{Lagrangian}}$$

$$\mathcal{L}: \Psi \times \mathbb{R}_+ \to \mathbb{R}$$
$$(\psi, \lambda) \mapsto R(\psi) + \lambda S(\psi)$$

II.1- Problem formulation

EP-learning [2]: Efficient plug-in risk estimator

1- Plug-in efficient estimator of $\Delta\mu$

2- Minimize risk (1) for $\psi \in \Psi$

$$R: \psi \mapsto E[\psi(X)^2 - 2\psi(X) \cdot \Delta\mu(X)] \quad (1)$$

Add constraint: Guarantee that π does **not increase** the average probability of adverse event beyond $\alpha \in \left[0, \frac{1}{2}\right]$

$$S: \psi \mapsto E\left[\sigma_\beta \circ \psi(X) \cdot \Delta\nu(X)\right] - \alpha \quad (2)$$

$$\nu(A, X) = E[\xi | A, X]$$
$$\Delta\nu(X) = \nu(1, X) - \nu(0, X)$$

Policy optimization objective

Minimize $R(\psi)$ w.r.t $\psi \in \Psi$

s.t. $S(\psi) \leq 0$

$$\xrightarrow{\text{Lagrangian}}$$

$$\mathcal{L}: \Psi \times \mathbb{R}_+ \to \mathbb{R}$$
$$(\psi, \lambda) \mapsto R(\psi) + \boxed{\lambda} S(\psi)$$

Parameters to fine-tune:

- β policy definition
- $\boxed{\lambda}$ weight given to constraint

II.2- Algorithm: EP-learner for constrained policy estimation

1- Estimate J cross-validated nuisance parameters: $\hat{\mu}_{n,j}, \hat{\nu}_{n,j}$

for $\boldsymbol{j}$ **= 1, … , J-1 do**:

for $\boldsymbol{\beta} \in \boldsymbol{B}$ **do**:

for $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$ **do**:

2.1- Plug-in $\hat{\mu}_{n,j}, \hat{\nu}_{n,j}$ in the objective function: $\hat{\mathcal{L}}_{n,j}(\psi, \lambda) \mapsto \hat{R}_{n,j}(\psi) + \lambda \hat{S}_{n,j}(\psi)$ **(4)**

2.2- Obtain minimizer of **(4)**: $\hat{\psi}_{n,j} \in \text{argmin}\{\hat{\mathcal{L}}_{n,j}(\psi, \lambda) \colon \psi \in \Psi\}$ [**Frank-Wolfe algorithm,** 3]

3- Identify the optimal β^{opt} and λ^{opt}

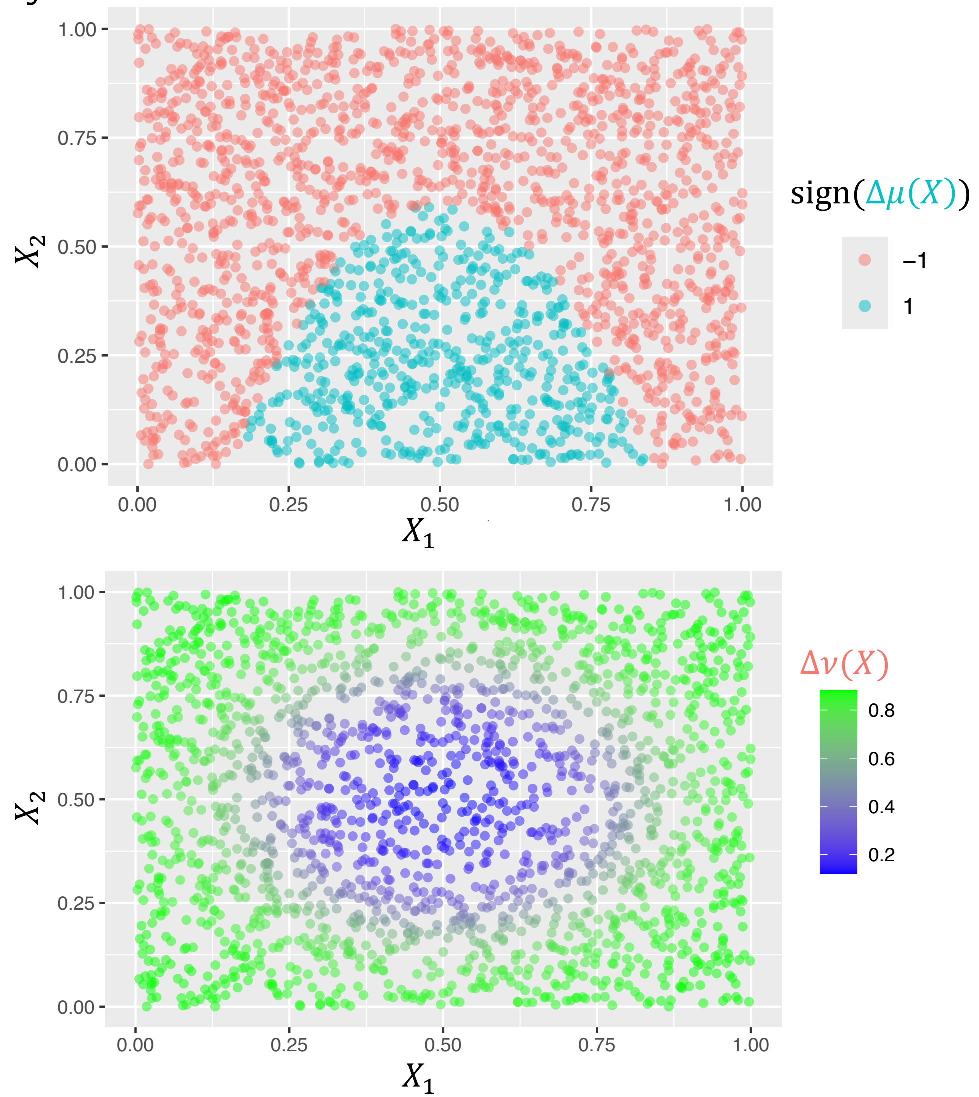
4- Obtain $\hat{\psi}_n$ minimizer of $\hat{\mathcal{L}}_{n,J}(\psi, \lambda^{opt})$

return $\hat{\pi}(X) = \sigma_{\beta^{opt}} \circ \hat{\psi}_n(X)$

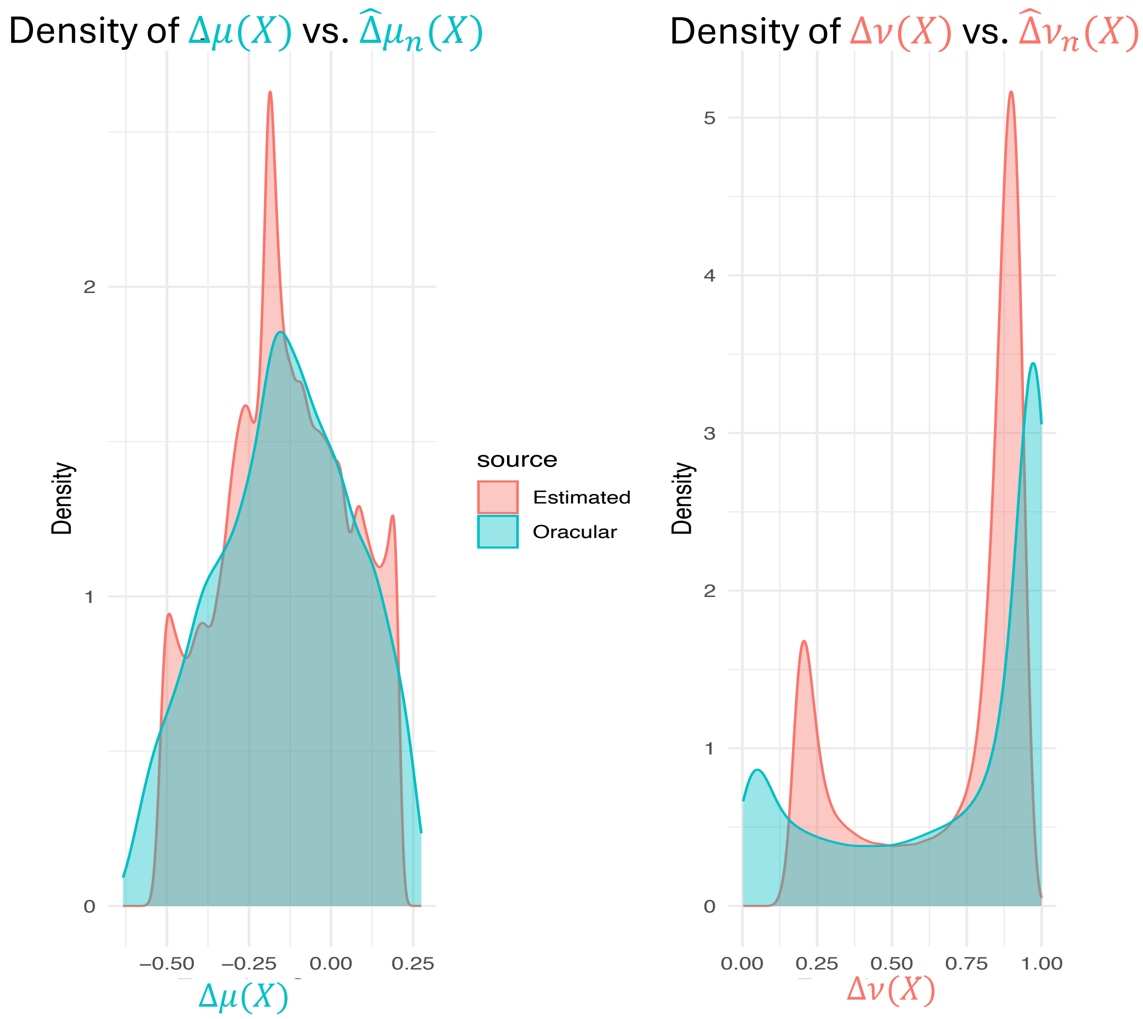
III. Results

III.1- Results: synthetic data

Synthetic data

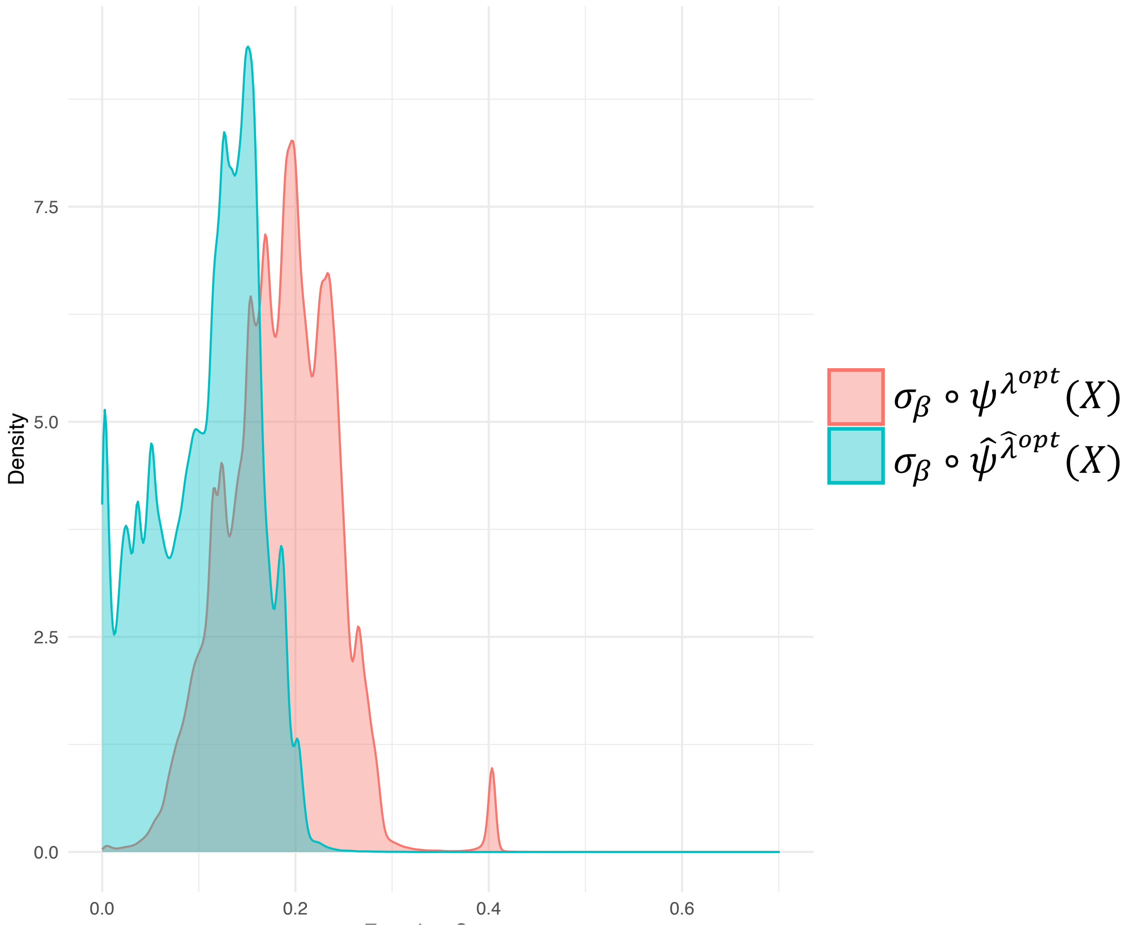


Nuisance parameter estimation

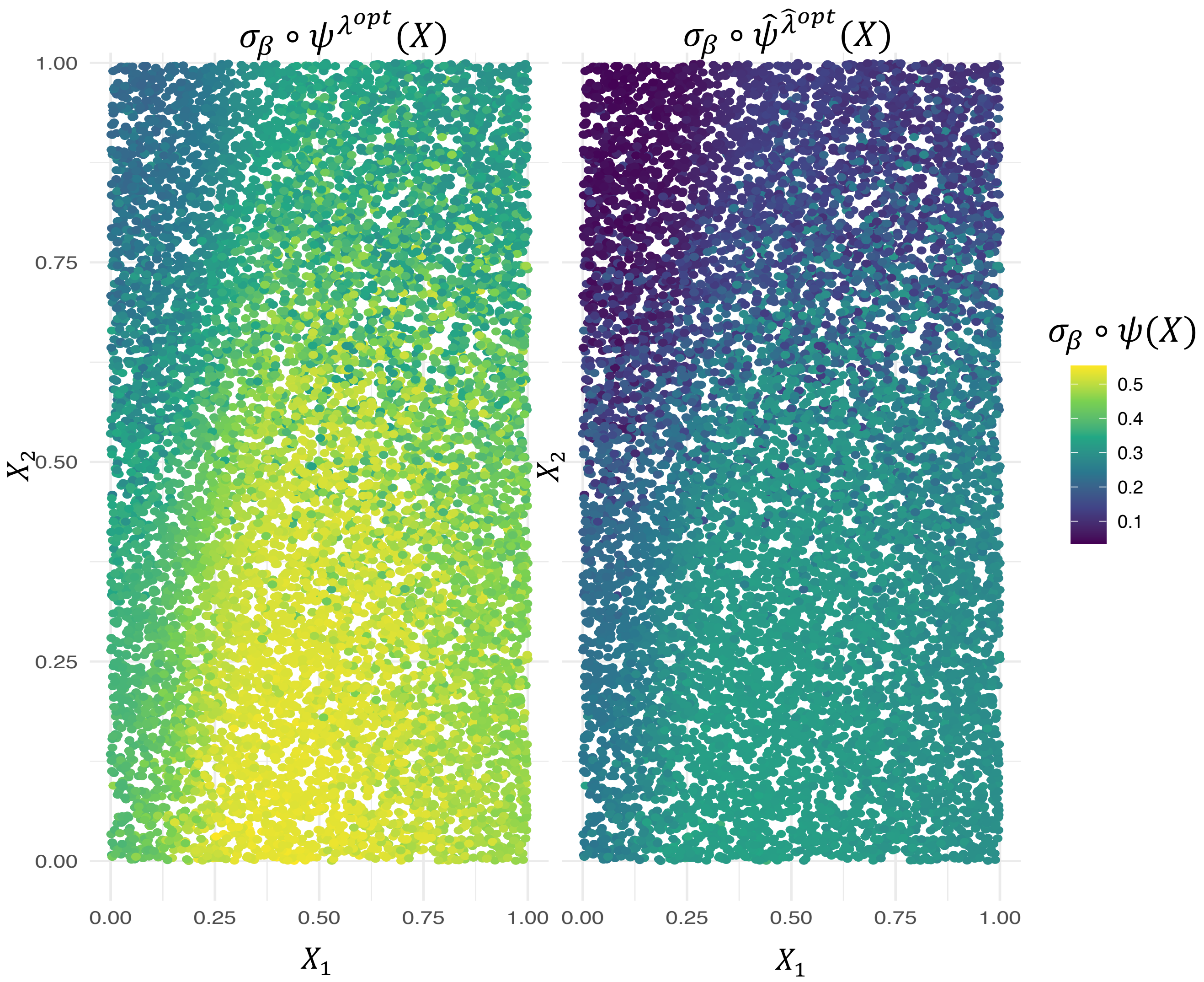


III.1- Results: synthetic data

Density comparison



Mean treatment probability comparison



IV. Discussion

Discussion

- Formulated a policy optimization problem with constraints
- Defined a strongly convex policy in a probabilistic style (in [0, 1])
- Adapted the EP-learner for multiple outcomes
- Optimizing policy in a convex function space using the Frank-Wolfe algorithm

Perspectives:

- Correct plug-in $\hat{\mathcal{L}}_n(\psi)$, while **preserving strong-convexity**
- **Alternated optimization**-estimator correction
- Compare to state of the art constrained policy optimization techniques

Thank you!

References

[1] Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. Journal of the American Statistical Association, 100(469):322–331.

[2] Van der Laan, L., Carone, M., and Luedtke, A. (2024). Combining T-learning and DR-learning: a framework for oracle-efficient estimation of causal contrasts arXiv preprint arXiv:2402.01972.

[3] Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In International conference on machine learning, pages 427–435. PMLR.

Annexes

Bias of objective function estimator

Minimize $\mathcal{L} : \psi \mapsto E\left[\psi(X)^2 - 2\psi(X) \cdot \Delta\mu(X) + \lambda\sigma_\beta \circ \psi(X) \cdot \Delta\nu(X)\right] - \lambda\alpha$

μ, ν unknown

Solution

Plug-in estimator: $\hat{\mathcal{L}}_n : \psi \mapsto E_{P_n}\left[\psi(X)^2 - 2\psi(X) \cdot \widehat{\Delta\mu}_n(X) + \lambda\sigma_\beta \circ \psi(X) \cdot \widehat{\Delta\nu}_n(X)\right] - \lambda\alpha$

Bias of objective function estimator

Minimize $\mathcal{L} : \psi \mapsto E\left[\psi(X)^2 - 2\psi(X) \cdot \Delta\mu(X) + \lambda\sigma_\beta \circ \psi(X) \cdot \Delta\nu(X)\right] - \lambda\alpha$

✗ μ, ν unknown

Solution

Plug-in estimator: $\hat{\mathcal{L}}_n : \psi \mapsto E_{P_n}\left[\psi(X)^2 - 2\psi(X) \cdot \widehat{\Delta\mu_n}(X) + \lambda\sigma_\beta \circ \psi(X) \cdot \widehat{\Delta\nu_n}(X)\right] - \lambda\alpha$

Evaluation via EIC derivation of $\mathcal{F} : P \mapsto \mathcal{L}_P(\psi)$

$$EIC_{\mathcal{F}}(P)(X, A, Y, \xi) = \nabla_1\mathcal{F}(P) + \underbrace{\nabla_2\mathcal{F}(P)}_{E[\nabla_2\mathcal{F}(P)] = 0}$$

Estimator bias

$$E[\nabla_1\mathcal{F}(P)(X, A, Y, \xi)] = E\left[\frac{2A - 1}{P(A|X)}\Big(-2\psi(X) \cdot \big(Y - \mu(A, X)\big) + \lambda\sigma_\beta \circ \psi(X) \cdot \big(\xi - \nu(A, X)\big)\Big)\right]$$

✗ Non-convex function of ψ

II.2- Algorithm: Alternated procedure

1- Estimate J cross-validated nuisance parameters: $\hat{\mu}_{n,j}, \hat{\nu}_{n,j}$

for $\boldsymbol{j} = \mathbf{1}, \ldots, \boldsymbol{J}\text{-}\mathbf{1}$ **do**:

for $\boldsymbol{\beta} \in \boldsymbol{B}$ **do**:

for $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$ **do**:

2- Run iteratively alternated procedure for fixed β, λ:

- **Minimize** $\hat{\mathcal{L}}^k_{n,j}$: $\hat{\psi}^k_{n,j} \in \operatorname{argmin}\{\hat{\mathcal{L}}^k_{n,j}(\psi, \lambda) \colon \psi \in \Psi\}$
- **Correct** $\hat{\mathcal{L}}^k_{n,j}$ by adjusting nuissance parameters $\hat{\mu}_{n,j}, \hat{\nu}_{n,j}$ for fixed $\hat{\psi}^k_{n,j}$

3- Identify the optimal β^{opt} and λ^{opt}

4- Obtain $\hat{\psi}_n$ minimizer of $\hat{\mathcal{L}}^k_{n,J}(\psi, \lambda^{opt})$

return $\hat{\pi}(X) = \sigma_{\beta^{opt}} \circ \hat{\psi}_n(X)$

II.2- Algorithm: Alternated procedure

Require: λ, β, $\hat{\mu}_{n,j}$, $\hat{\nu}_{n,j}$, $\boldsymbol{e_n}$, $\boldsymbol{\gamma}$, **max_iter**

Initialization:

Optimization: $\boldsymbol{\hat{\mu}^0_{n,j}(\epsilon_1)} = \hat{\mu}_{n,j}$, $\boldsymbol{\hat{\nu}^0_{n,j}(\epsilon_2)} = \hat{\nu}_{n,j}$

$\psi^0_{n,j} \in \operatorname{argmin}\{\hat{\mathcal{L}}^0_{n,j}(\psi,\lambda) : \psi \in \Psi\}$ [**Frank-Wolfe algorithm, 3**]

While $\left(\sum(\hat{\psi}^{k-1}_{n,j} - \hat{\psi}^k_{n,j})^2\right) < \gamma$ & $k <$ max_iter:

$k = k + 1$

Correction: $\epsilon \in \operatorname{argmin} \ell^k_n(\epsilon) = \begin{cases} \frac{-1}{n} \sum_{i=1}^n Y_i \log(\hat{\mu}^k_n(\epsilon_1)) + (1 - Y_i)\log(1 - \hat{\mu}^k_n(\epsilon_1)) \\ \frac{-1}{n} \sum_{i=1}^n \xi_i \log(\hat{\nu}^k_n(\epsilon_2)) + (1 - \xi_i)\log(1 - \hat{\nu}^k_n(\epsilon_2)) \end{cases}$

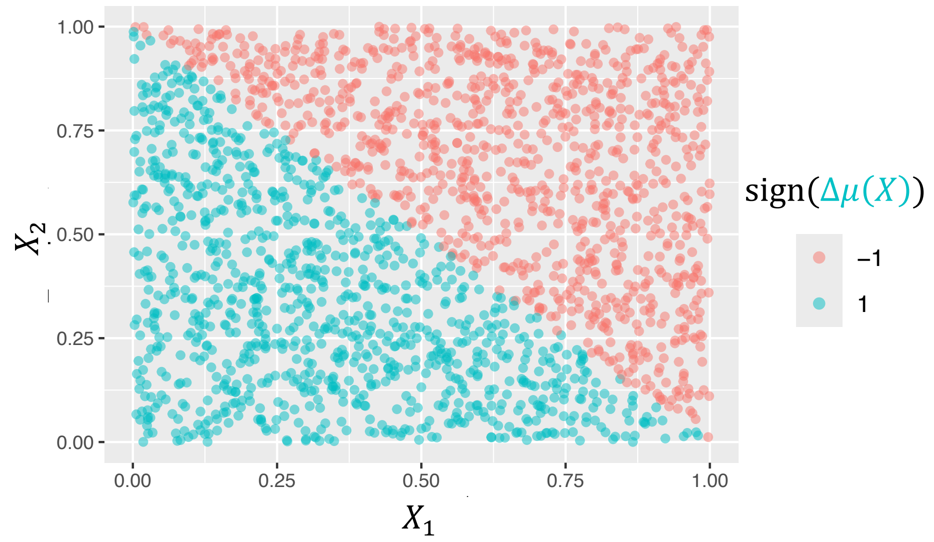
Nuisance parameter update:

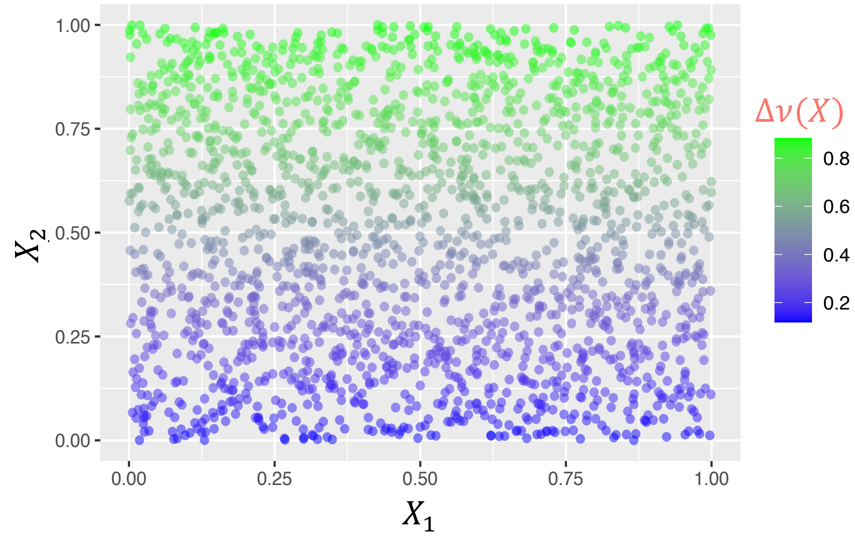
$\hat{\mu}^k_{n,j}(\epsilon_1) = \operatorname{expit}(\operatorname{logit}(\hat{\mu}^0_{n,j}) + \frac{2A-1}{e_n(A,X)} \sum \epsilon_{1,l} \hat{\psi}_l(X))$ $\quad \hat{\nu}^k_{n,j}(\epsilon_2) = \operatorname{expit}(\operatorname{logit}(\hat{\nu}^0_{n,j}) + \frac{2A-1}{e_n(A,X)} \sum \epsilon_{2,l} \sigma_\beta \circ \hat{\psi}_l(X))$

Optimization: $\hat{\psi}^k_{n,j} \in \operatorname{argmin}\{\hat{\mathcal{L}}^k_{n,j}(\psi,\lambda) : \psi \in \Psi\}$ [**Frank-Wolfe algorithm, 3**]

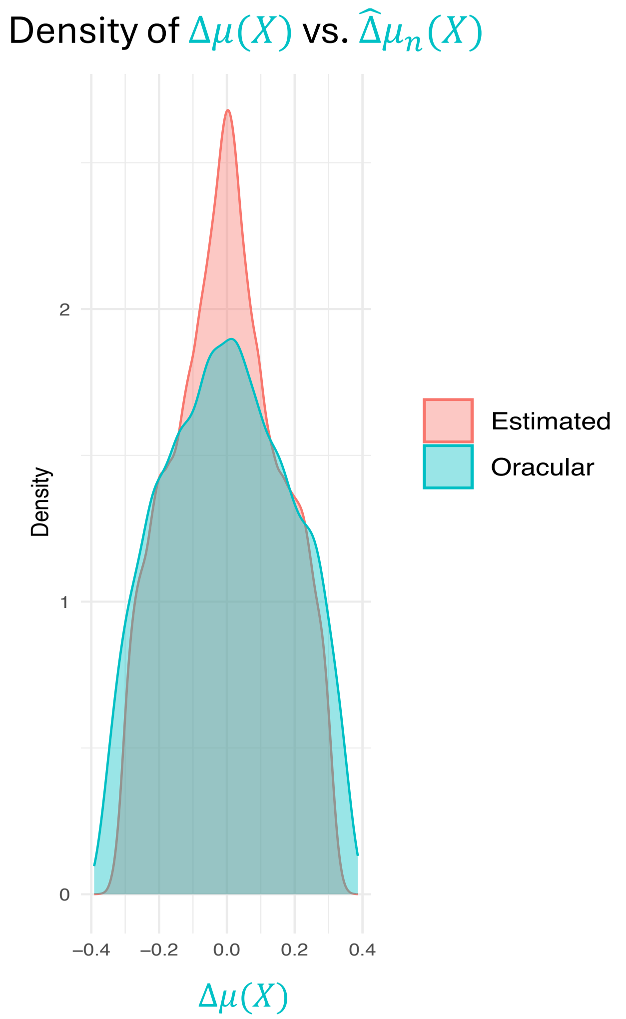
III.2- Results: synthetic data, second scenario

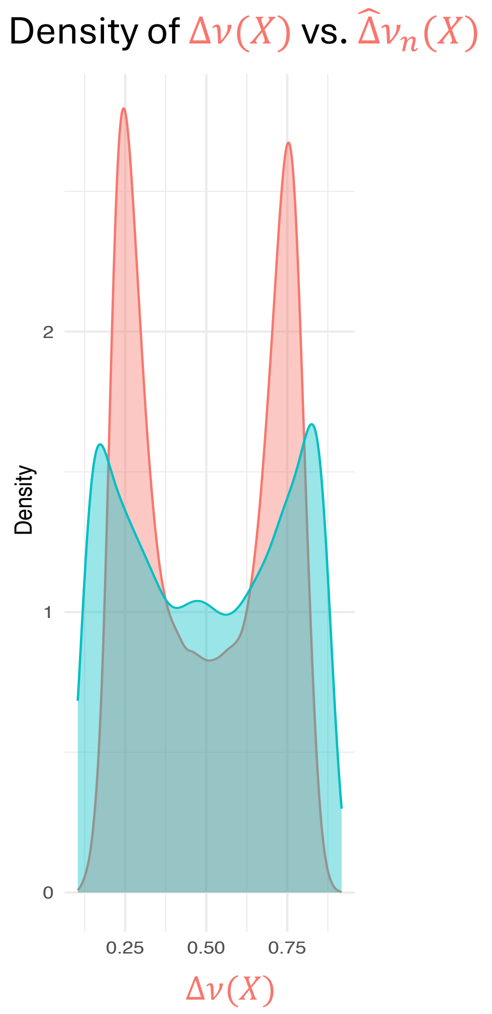
Synthetic data





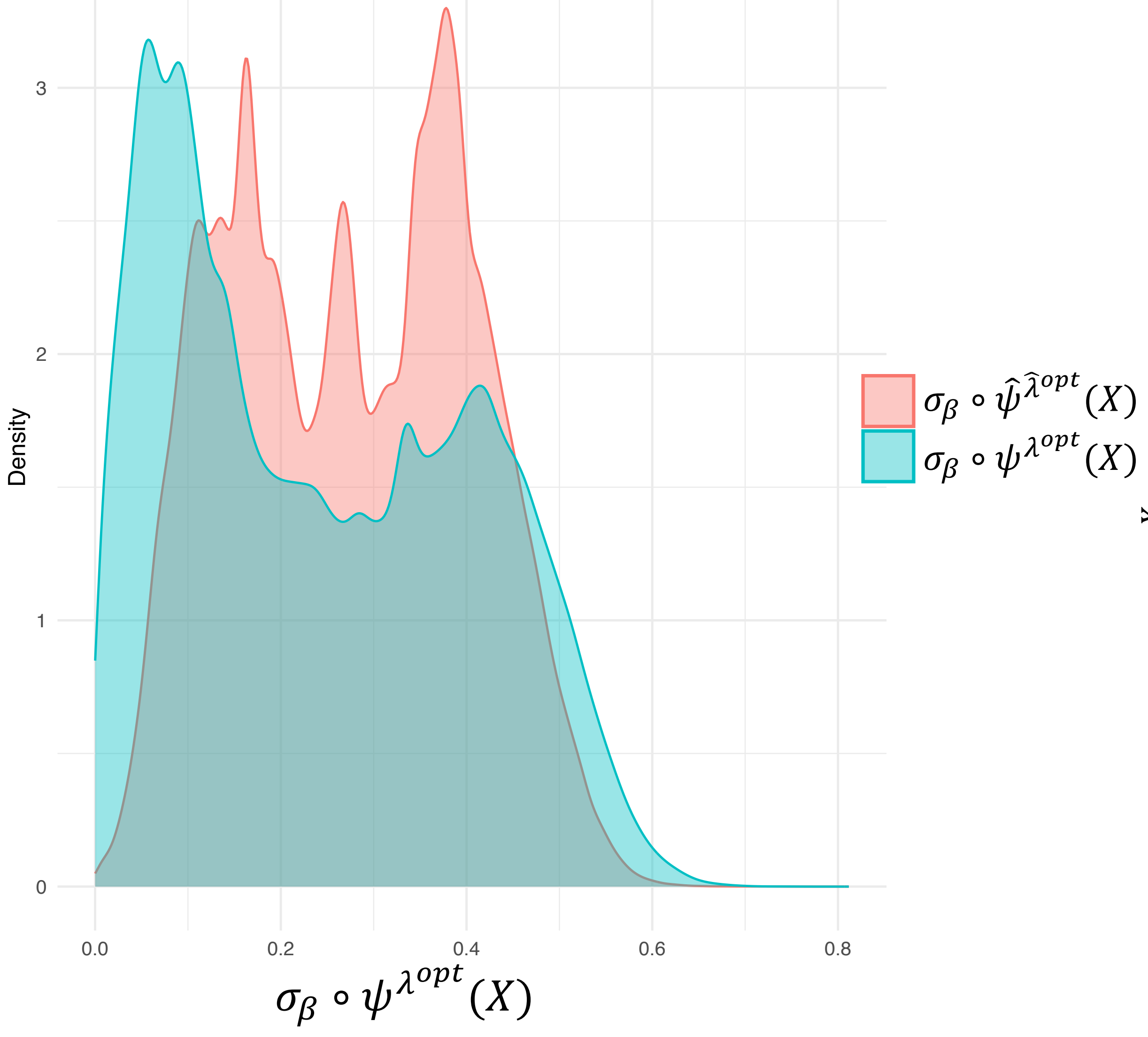
Nuisance parameter estimation





III.2- Results: synthetic data, second scenario

Density comparison



Mean treatment probability comparison