Targeted learning with right-censored data using the state learner

Anders Munch
joint work with Thomas Gerds

Section of Biostatistics, UCPH

March 20, 2025

Targeted or debiased machine learning*

Low-dimensional target parameter

$$\Psi\colon \mathcal{P} \to \mathbb{R}, \quad \text{such that} \quad \Psi(P) = \tilde{\Psi}(\nu_1(P), \ldots, \nu_B(P)),$$

for high-dimensional nuisance parameters $\nu_1, \ldots, \nu_B$.

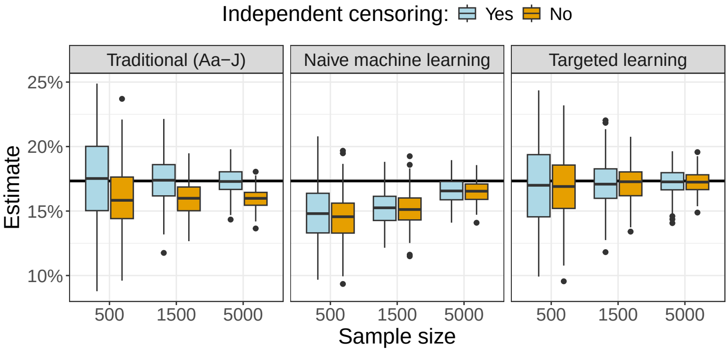
*[van der Laan and Rose, 2011, Chernozhukov et al., 2018]

Targeted or debiased machine learning*

Low-dimensional target parameter

$$\Psi \colon \mathcal{P} \to \mathbb{R}, \quad \text{such that} \quad \Psi(P) = \tilde{\Psi}(\nu_1(P), \ldots, \nu_B(P)),$$

for high-dimensional nuisance parameters $\nu_1, \ldots, \nu_B$.



Adapted from Figure 4 in [Munch et al., 2023].

*[van der Laan and Rose, 2011, Chernozhukov et al., 2018]

Examples of estimands

Available data $O = (X, \tilde{T}, \tilde{D}) \sim P \in \mathcal{P}$, where

$$
\begin{aligned}
&X = W \in \mathbb{R}^d \text{ or } X = (W, A) \in \mathbb{R}^d \times \{0, 1\}, \\
&\tilde{T} = T \wedge C \\
&\tilde{D} = \mathbb{1}\{T \leq C\}D, \quad D = 1 \text{ or } D \in \{1, 2\}.
\end{aligned}
$$

The variables T and D are the uncensored event time and event indicator.

Examples of estimands

Available data $O = (X, \tilde{T}, \tilde{D}) \sim P \in \mathcal{P}$, where

$$X = W \in \mathbb{R}^d \text{ or } X = (W, A) \in \mathbb{R}^d \times \{0, 1\},$$
$$\tilde{T} = T \wedge C$$
$$\tilde{D} = \mathbb{1}\{T \leq C\}D, \quad D = 1 \text{ or } D \in \{1, 2\}.$$

The variables T and D are the uncensored event time and event indicator.

$$Q(T > t) \overset{(!)}{=} \mathbb{E}_P[e^{-\Lambda_P(t|X)}], \quad (T, X) \sim Q,$$

where

$$\Lambda_P(\mathrm{d}t \mid x) = P(\tilde{T} \in \mathrm{d}t, \tilde{D} = 1 \mid \tilde{T} \geq t, X = x).$$

For (!) to hold we need $C \perp\!\!\!\perp T \mid X$.

Examples of estimands

Available data $O = (X, \tilde{T}, \tilde{D}) \sim P \in \mathcal{P}$, where

$$X = W \in \mathbb{R}^d \text{ or } X = (W, A) \in \mathbb{R}^d \times \{0, 1\},$$
$$\tilde{T} = T \wedge C$$
$$\tilde{D} = \mathbb{1}\{T \leq C\}D, \quad D = 1 \text{ or } D \in \{1, 2\}.$$

The variables T and D are the uncensored event time and event indicator.

Use $\{(T^a, D^a) : a \in \{0, 1\}\} \sim Q$, to denote potential outcomes.

$$Q(T^a \leq t, D^a = 1) \overset{(!)}{=} \mathbb{E}_P\left[\int_0^t e^{-[\Lambda_{1,P}+\Lambda_{2,P}](u \mid a, W)} \Lambda_{1,P}(\mathrm{d}u \mid a, W)\right],$$

$$\Lambda_{d,P}(\mathrm{d}t \mid a, w) = P(\tilde{T} \in \mathrm{d}t, \tilde{D} = d \mid \tilde{T} \geq t, A = a, W = w).$$

For (!) to hold we need both $C \perp\!\!\!\perp T \mid X$ and causal assumptions.

Targeted learning with super learning

$$\Psi(P) = \mathbb{E}_P\left[\int_0^t e^{-[\Lambda_{1,P}+\Lambda_{2,P}](u|a,W)}\Lambda_{1,P}(\mathrm{d}u \mid a, W)\right]$$

can be written as

$$\Psi(P) = \tilde{\Psi}(\Lambda_{1,P}, \Lambda_{2,P}, P_W).$$

$$\tilde{\Psi}(\hat{\Lambda}_{1,n}, \hat{\Lambda}_{2,n}, \mathbb{P}_n) \quad + \quad \begin{array}{c}\text{Targeting/debiasing step} \\ \text{using } \hat{\Gamma}_n \text{ and } \hat{\pi}_n\end{array} \longrightarrow \quad \hat{\Psi}_n^*$$

Asymptotic valid inference and $O_P(n^{-1/2})$ convergence for $\hat{\Psi}_n^*$ if

$$\|\hat{\nu}_n - \nu\|_{P,2} = o_P(n^{-1/4}), \quad \text{for all } \hat{\nu}_n \in \{\hat{\Lambda}_{1,n}, \hat{\Lambda}_{2,n}, \hat{\Gamma}_n, \hat{\pi}_n\}.$$

Targeted learning with super learning

$$\Psi(P) = \mathbb{E}_P\left[\int_0^t e^{-[\Lambda_{1,P}+\Lambda_{2,P}](u \mid a,W)}\Lambda_{1,P}(\mathrm{d}u \mid a, W)\right]$$

can be written as

$$\Psi(P) = \tilde{\Psi}(\Lambda_{1,P}, \Lambda_{2,P}, P_W).$$

$$\tilde{\Psi}(\hat{\Lambda}_{1,n}, \hat{\Lambda}_{2,n}, \mathbb{P}_n) \quad + \quad \begin{array}{c}\text{Targeting/debiasing step}\\ \text{using } \hat{\Gamma}_n \text{ and } \hat{\pi}_n\end{array} \longrightarrow \quad \hat{\Psi}_n^*$$

Asymptotic valid inference and $O_P(n^{-1/2})$ convergence for $\hat{\Psi}_n^*$ if

$$\|\hat{\nu}_n - \nu\|_{P,2} = o_P(n^{-1/4}), \quad \text{for all } \hat{\nu}_n \in \{\hat{\Lambda}_{1,n}, \hat{\Lambda}_{2,n}, \hat{\Gamma}_n, \hat{\pi}_n\}.$$

Attempt to achieve this using a super learner.

Super learning[†]

$$\mathcal{D}_n \quad = \quad \boxed{\begin{array}{c|c|ccc} O_1 & O_2 & \quad\cdots\quad & \quad\cdots\quad & \quad\cdots\quad \end{array}}$$

$$\begin{aligned} \text{Learner} \quad & \mathcal{D}_n \longmapsto a(\mathcal{D}_n) = \hat{\nu}_n \\ \text{Library} \quad & \mathcal{A} = \{a_1, a_2, \dots, a_M\} \\ \text{Loss function} \quad & \mathcal{V} \times \mathcal{O} \ni (\nu, O) \longmapsto L(\nu, O) \in \mathbb{R} \end{aligned}$$

$$\text{Discrete SL} = \hat{a}_n = \operatorname*{argmin}_{a \in \mathcal{A}} \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{D}_n^k|} \sum_{O_i \in \mathcal{D}_n^k} L\big(a(\mathcal{D}_n^{-k}), O_i\big),$$

[†] Stone [1974], Geisser [1975], Wolpert [1992], Breiman [1996], van der Laan et al. [2007]

Super learning[†]

$$\mathcal{D}_n^{-1} = \qquad \boxed{\quad \cdots \quad} \boxed{\quad \cdots \quad}$$

$$\mathcal{D}_n^{1} = \boxed{O_1 \,|\, O_2 \,|\quad \cdots \quad}$$

Learner $\mathcal{D}_n \longmapsto a(\mathcal{D}_n) = \hat{\nu}_n$

Library $\mathcal{A} = \{a_1, a_2, \ldots, a_M\}$

Loss function $\mathcal{V} \times \mathcal{O} \ni (\nu, O) \longmapsto L(\nu, O) \in \mathbb{R}$

$$\text{Discrete SL} = \hat{a}_n = \operatorname*{argmin}_{a \in \mathcal{A}} \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{D}_n^k|} \sum_{O_i \in \mathcal{D}_n^k} L\big(a(\mathcal{D}_n^{-k}), O_i\big),$$

[†]Stone [1974], Geisser [1975], Wolpert [1992], Breiman [1996], van der Laan et al. [2007]

Super learning[†]

$$\mathcal{D}_n^{-2} = \boxed{O_1 \,|\, O_2 \,|\, \quad \cdots \quad} \qquad\qquad \boxed{\quad \cdots \quad}$$

$$\mathcal{D}_n^{2} = \qquad\qquad \boxed{\quad \cdots \quad}$$

Learner $\mathcal{D}_n \longmapsto a(\mathcal{D}_n) = \hat{\nu}_n$

Library $\mathcal{A} = \{a_1, a_2, \dots, a_M\}$

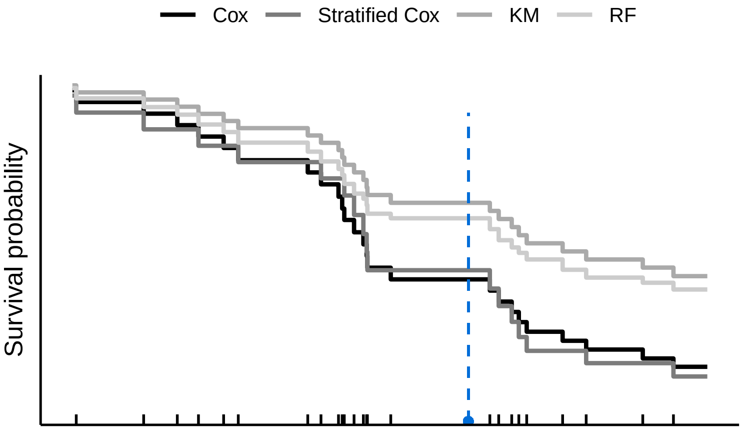
Loss function $\mathcal{V} \times \mathcal{O} \ni (\nu, O) \longmapsto L(\nu, O) \in \mathbb{R}$

$$\text{Discrete SL} = \hat{a}_n = \operatorname*{argmin}_{a \in \mathcal{A}} \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{D}_n^k|} \sum_{O_i \in \mathcal{D}_n^k} L\big(a(\mathcal{D}_n^{-k}), O_i\big),$$

[†] Stone [1974], Geisser [1975], Wolpert [1992], Breiman [1996], van der Laan et al. [2007]

Evaluating performance in hold-out folds is a challenge with right-censored data

The negative log-likelihood is unsuited for super learning

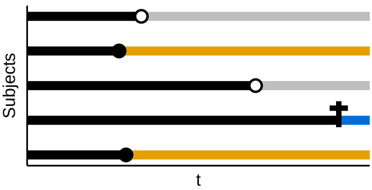


IPCW and pseudo-outcome require pre-specified censoring estimator

$$\begin{tikzcd}
\widehat{S} \arrow[d, bend right] & \widehat{W_G} \arrow[l, bend right] \\
\widehat{W_S} \arrow[r, bend right] & \widehat{G} \arrow[u, bend right]
\end{tikzcd}$$

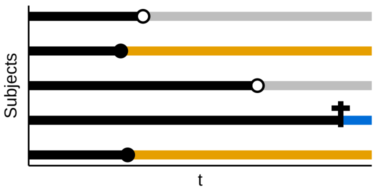
The state learner: Model all 'states' of the *observed* data

$$N(t) = \mathbb{1}\{\tilde{T} \leq t, \tilde{D} = 1\} + 2\,\mathbb{1}\{\tilde{T} \leq t, \tilde{D} = 2\} - \mathbb{1}\{\tilde{T} \leq t, \tilde{D} = 0\}$$



The state learner: Model all 'states' of the *observed* data

$$N(t) = \mathbb{1}\{\tilde{T} \le t, \tilde{D} = 1\} + 2\,\mathbb{1}\{\tilde{T} \le t, \tilde{D} = 2\} - \mathbb{1}\{\tilde{T} \le t, \tilde{D} = 0\}$$

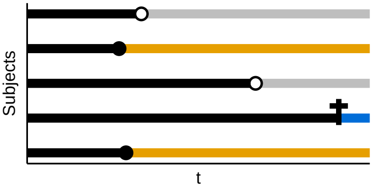


Build a super learner for the function

$$F(t, k, w, a) = P(N(t) = k \mid W = w, A = a).$$

The state learner: Model all 'states' of the *observed* data

$$N(t) = \mathbb{1}\{\tilde{T} \leq t, \tilde{D} = 1\} + 2\,\mathbb{1}\{\tilde{T} \leq t, \tilde{D} = 2\} - \mathbb{1}\{\tilde{T} \leq t, \tilde{D} = 0\}$$



Build a super learner for the function

$$F(t, k, w, a) = P(N(t) = k \mid W = w, A = a).$$

$$L(F, O) = \int_0^\tau \sum_{j=-1}^{2} \Big(F(t, j, W, A) - \mathbb{1}\{N(t) = j\}\Big)^2 \,\mathrm{d}t.$$

Building a library for F

The formulas

$$F(t,0,w,a) = \prod_{0}^{t} \left(1 - [\Lambda_1 + \Lambda_2 + \Gamma]\,(\mathrm{d}s \mid w,a)\right),$$
$$F(t,j,w,a) = \int_0^t F(t-,0,w,a)\Lambda_j(\mathrm{d}s \mid w,a), \quad j \in \{1,2\},$$
$$F(t,-1,w,a) = \int_0^t F(t-,0,w,a)\Gamma(\mathrm{d}s \mid w,a),$$

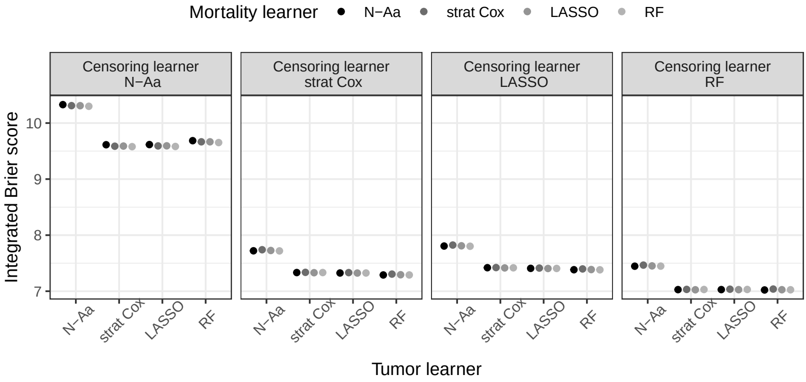
can be used to build a library for F from libraries for Λ_1, Λ_2, Γ,

$$\mathcal{F}(\mathcal{A},\mathcal{B},\mathcal{C}) = \{F_{a,b,c} : a \in \mathcal{A}, b \in \mathcal{B}, c \in \mathcal{C}\}.$$

Similarly, we can obtain Λ_j and Γ from F.

Illustration with data from a prostate cancer study‡

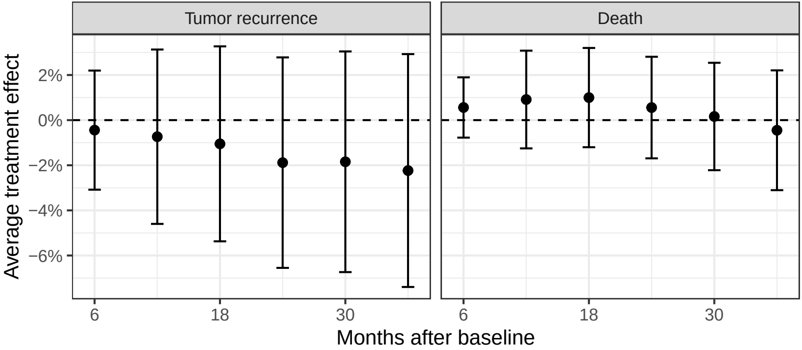
The learners selected by the state learner.



‡Kattan et al. [2000].

Illustration with data from a prostate cancer study‡

The estimated average treatment effect of hormone therapy.



‡Kattan et al. [2000].

Additional details

Munch and Gerds [2024] provide

- a consistency result
- a finite sample oracle inequality
- details for the targeting step

Prototype available at https://github.com/amnudn/statelearner.

Thank you for the attention!

References

L. Breiman. Stacked regressions. *Machine learning*, 24(1):49–64, 1996.

V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 2018. doi: 10.1111/ectj.12097.

S. Geisser. The predictive sample reuse method with applications. *Journal of the American statistical Association*, 70(350):320–328, 1975.

M. W. Kattan, M. J. Zelefsky, P. A. Kupelian, P. T. Scardino, Z. Fuks, and S. A. Leibel. Pretreatment nomogram for predicting the outcome of three-dimensional conformal radiotherapy in prostate cancer. *Journal of clinical oncology*, 2000.

A. Munch and T. A. Gerds. The state learner–a super learner for right-censored data. *arXiv preprint arXiv:2405.17259*, 2024.

A. Munch, M. S. Breum, T. Martinussen, and T. A. Gerds. Targeted estimation of state occupation probabilities for the non-markov illness-death model. *Scandinavian Journal of Statistics*, 50(3):1532–1551, 2023.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2):111–133, 1974.

M. J. van der Laan and S. Rose. *Targeted learning: causal inference for observational and experimental data*. Springer Science & Business Media, 2011.

M. J. van der Laan, E. C. Polley, and A. E. Hubbard. Super learner. *Statistical applications in genetics and molecular biology*, 6(1), 2007.

D. H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.