The state learner
a super learner for right-censored data

Anders Munch
joint work with Thomas Gerds

January 30, 2024

Setting and motivation

Construct a targeted estimator of a low-dimensional target parameter such as the average treatment effect.

Available data are right-censored.

Requires estimation of high-dimensional nuisance parameters such as the conditional survival and censoring mechanism.

Use super learning to alleviate the need to fully pre-specify estimators of the nuisance parameters.

Challenge: Existing super learners restrict which learners can be included in the library or require pre-specification of an estimator of the censoring mechanism.

Super learning†

Need to estimate the parameter

$$f = \operatorname*{argmin}_{f \in \mathcal{F}} P[L(f, \cdot)],$$

using the data set $\mathcal{D}_n = \{O_1, \dots, O_n\}$.

Learner	algorithm a that produces estimates, $\mathcal{D}_n \mapsto a(\mathcal{D}_n) = \hat{f}_n$
Library	collection of learners, $\mathcal{A} = \{a_1, a_2, \dots, a_M\}$
Loss function	$L\colon \mathcal{F} \times \mathcal{O} \to \mathbb{R}$

Combine learners from the library into a new learner with performance almost as good as the best of them.

Regression example: $L(f, O) = (f(X) - Y)^2$, library consiting of the learners `lm`, `glm`, `glmnet`, `rfsrc`, ...

---

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

Discrete super learner

Learner algorithm a that produces estimates, $\mathcal{D}_n \mapsto a(\mathcal{D}_n) = \hat{f}_n$

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

Loss function $L \colon \mathcal{F} \times \mathcal{O} \to \mathbb{R}$

The discrete super learning is the data-adaptive learner

$$\hat{a}_n = \operatorname*{argmin}_{a \in \mathcal{A}} \hat{R}_n(a; L),$$

where

$$\hat{R}_n(a; L) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{D}_n^k|} \sum_{O_i \in \mathcal{D}_n^k} L\Big(a(\mathcal{D}_n^{-k}), O_i\Big), \quad \text{with} \quad \mathcal{D}_n^{-k} = \mathcal{D}_n \setminus \mathcal{D}_n^k.$$

Right-censored data

Notation

- X vector of baseline covariates
- T time to event variable, $T > 0$
- C censoring time, $C > 0$
- Q distribution of the data of interest $(X, T) \sim Q$
- $\tilde{T}$ censored time to event variable, $\tilde{T} = \min(T, C)$
- Δ binary event indicator, $\Delta = \mathbb{1}\{T \leq C\}$
- P distribution of the observed data, $O = (X, \tilde{T}, \Delta) \sim P$

We use Λ and Γ to denote the conditional cumulative hazard functions for T and C, i.e.,

$$\Lambda(\mathrm{d}t \mid x) = Q(T \in \mathrm{d}t \mid T \geq t, X = x).$$

Assuming $T \perp\!\!\!\perp C \mid X$ and positivity so that Λ and Γ are identifiable from P on some $[0, \tau]$.

Super learning and targeted learning

Parameters of interest could be

$$\Psi_t(Q) = Q(T > t),$$
$$\Psi_t(Q) = \mathbb{E}_Q[Q(T > t \mid X, A = 1) - Q(T > t \mid X, A = 0)], \quad \text{or}$$
$$\Psi_t(Q) = \mathbb{E}_Q[\mathbb{E}_Q[T \wedge t \mid X, A = 1] - \mathbb{E}_Q[T \wedge t \mid X, A = 0]].$$

Can be expressed using Λ so we they are identifiable from $P \in \mathcal{P}$ and we may write $\tilde{\Psi}_t(P) = \Psi_t(Q(P))$.

The efficient influence function for $\tilde{\Psi}_t$ can be indexed by (Λ, Γ) or (Λ, Γ, π), where π is the propensity model.

Goal is to use super learning to estimate the parameters (Λ, Γ) so that we can construct a targeted estimator of $\tilde{\Psi}_t$.

Super learning with right-censored data

Q distribution of $(X, T) \sim Q$

P distribution of $O = (X, \tilde{T}, \Delta) \sim P$

Typically want to estimate a feature of Q such as Λ.

Data set $\mathcal{D}_n = \{O_1, \dots, O_n\}$, with $O = (X, \tilde{T}, \Delta)$ from some P available.

Super learning relies on calculating

$$\hat{R}_n(a; L) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{D}_n^k|} \sum_{O_i \in \mathcal{D}_n^k} L\big(a(\mathcal{D}_n^{-k}), O_i\big), \quad \text{with} \quad \mathcal{D}_n^{-k} = \mathcal{D}_n \setminus \mathcal{D}_n^k.$$

Super learning with right-censored data

Q distribution of $(X, T) \sim Q$
P distribution of $O = (X, \tilde{T}, \Delta) \sim P$

Typically want to estimate a feature of Q such as Λ.

Data set $\mathcal{D}_n = \{O_1, \dots, O_n\}$, with $O = (X, \tilde{T}, \Delta)$ from some P available.

Super learning relies on calculating

$$\hat{R}_n(a; L) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{D}_n^k|} \sum_{O_i \in \mathcal{D}_n^k} L\big(a(\mathcal{D}_n^{-k}), O_i\big), \quad \text{with} \quad \mathcal{D}_n^{-k} = \mathcal{D}_n \setminus \mathcal{D}_n^k.$$

$a(\mathcal{D}_n^{-k})$ Training learners in censored data ✓

$L(a(\mathcal{D}_n^{-k}), O_i)$ Evaluating fitted learners in censored data ?

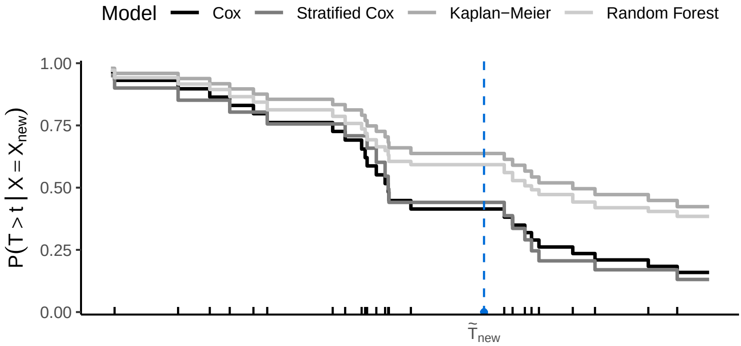
The negative log-likelihood is unsuited for super learning

Commonly used loss function for survival data is the negative (partial) log-likelihood.

Factorizes into separate components for Λ and Γ.

Unsuited for general use due to point masses.

The negative log-likelihood is unsuited for super learning



Existing approaches

Negative log-likelihood loss function (e.g., Polley and van der Laan [2011])

Requires discrete time or modeling of a Lebesgue hazard function.

Pseudo-observations (e.g., Sachs et al. [2019])

Requires pre-specification of an estimator of the censoring mechanism.

IPCW (e.g., Graf et al. [1999], Hothorn et al. [2006], Gerds and Schumacher [2006], Gonzalez Ginestet et al. [2021])

Requires pre-specification of an estimator of the censoring mechanism.

Iterative IPCW (Westling et al. [2021], Han et al. [2021])

Theoretical guarantees?

Proposal: Model the states of the *observed* data

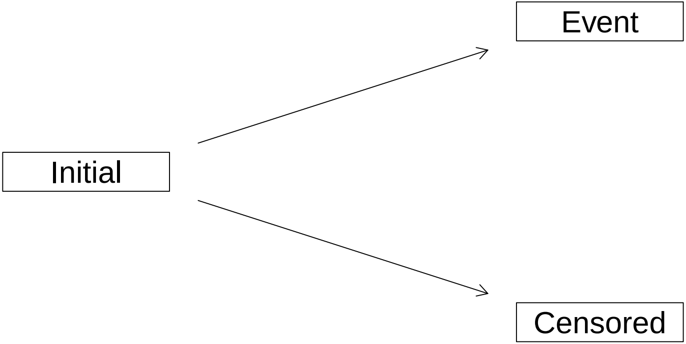
Proposal: Model the states of the *observed* data

$$(X, T) \sim Q$$

$$\boxed{\text{Initial}} \longrightarrow \boxed{\text{Event}}$$

Proposal: Model the states of the *observed* data

$$(X, \tilde{T}, \Delta) \sim P$$



Conditional state-occupation probabilities for observed data

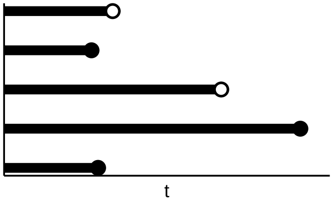
Define

$$\eta(t) = \mathbb{1}\{\tilde{T} \leq t, \Delta = 1\} + 2\,\mathbb{1}\{\tilde{T} \leq t, \Delta = 0\} \in \{0, 1, 2\},$$

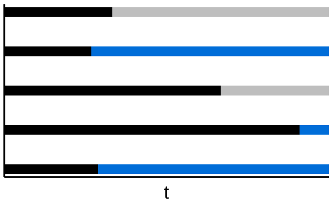
and

$$F(t, j, x) = P(\eta(t) = j \mid X = x), \quad \text{for all } t \geq 0,\ j \in \{0, 1, 2\},\ x \in \mathbb{R}^d.$$

$O = (X, \tilde{T}, \Delta)$



$O = (X, \{\eta(t) : t \geq 0\})$



The state learner

$$F(t, j, x) = P(\eta(t) = j \mid X = x), \quad \text{for all } t \geq 0,\ j \in \{0, 1, 2\},\ x \in \mathbb{R}^d.$$

The state learner is a super learner of F.

Performance can be evaluated using, e.g., the integrated Brier score $\bar{B}_\tau(F, O) = \int_0^\tau B_t(F, O)\,\mathrm{d}t$, where

$$B_t(F, O) = \sum_{j=0}^{2} (F(t, j, X) - \eta(t))^2.$$

Loss function does not depend on unknown nuisance parameters.

No modeling of Lebesgue hazards or densities required.

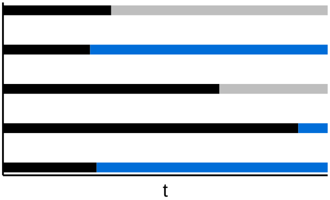
Expressing F using Λ and Γ

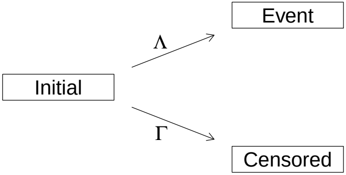
$$F(t,1,x) = P(\tilde{T} \le t, \Delta = 1 \mid X = x) = \int_0^t e^{-\Lambda(s|x)-\Gamma(s|x)}\Lambda(\mathrm{d}s \mid x),$$

$$F(t,2,x) = P(\tilde{T} \le t, \Delta = 0 \mid X = x) = \int_0^t e^{-\Lambda(s|x)-\Gamma(s|x)}\Gamma(\mathrm{d}s \mid x),$$

$$F(t,0,x) = P(\tilde{T} > t \mid X = x) = 1 - F(t,1,x) - F(t,2,x).$$

$\eta(t) : t \ge 0$





Constructing a library for learning F

Given libraries $\mathcal{A}$ and $\mathcal{B}$ for learning Λ and Γ, respectively, define

$$\mathcal{F}(\mathcal{A}, \mathcal{B}) = \{\varphi_{a,b} : a \in \mathcal{A}, b \in \mathcal{B}\},$$

where

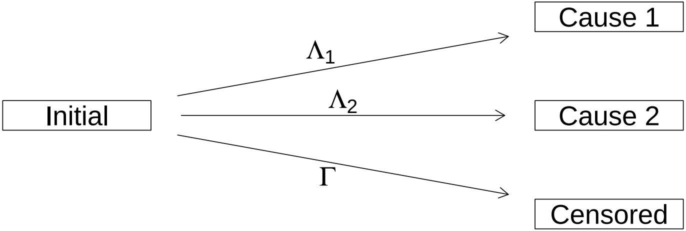
$$\varphi_{a,b}(\mathcal{D}_n)(t, 1, x) = \int_0^t e^{-a(\mathcal{D}_n)(s|x) - b(\mathcal{D}_n)(s|x)} a(\mathcal{D}_n)(\mathrm{d}s \mid x),$$

$$\dots$$

Evaluate performance of $\varphi_{a,b} \in \mathcal{F}(\mathcal{A}, \mathcal{B})$ as

$$\hat{R}_n(\varphi_{a,b}; \bar{B}_\tau) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{D}_n^k|} \sum_{O_i \in \mathcal{D}_n^k} \int_0^\tau \sum_{j=0}^{2} \left\{ \varphi_{a,b}(\mathcal{D}_n^{-k})(t, j, X_i) - \eta_i(t) \right\}^2 \mathrm{d}t.$$

Extension to competing risks setting



$$\eta(t) = \mathbb{1}\{\tilde{T} \leq t, \tilde{D} = 1\} + 2\,\mathbb{1}\{\tilde{T} \leq t, \tilde{D} = 2\} + 3\,\mathbb{1}\{\tilde{T} \leq t, \tilde{D} = 0\}.$$

$$F(t, 1, x) = P(\tilde{T} \leq t, \tilde{D} = 1 \mid X = x) = \int_0^t e^{-\Lambda_1(s|x) - \Lambda_2(s|x) - \Gamma(s|x)} \Lambda_1(\mathrm{d}s \mid x),$$

$$\ldots$$

$$\mathcal{F}(\mathcal{A}_1, \mathcal{A}_2, \mathcal{B}) = \{\varphi_{a_1, a_2, b} : a_1 \in \mathcal{A}_1, a_2 \in \mathcal{A}_2, b \in \mathcal{B}\},$$

Prototype[†]

```r
library(riskRegression)
data(Melanoma, package="riskRegression")
setDT(Melanoma)
head(Melanoma)
```

```
   time status                    event invasion ici      epicel       ulcer th
1:   10      2       death.other.causes  level.1   2     present     present  6
2:   30      2       death.other.causes  level.0   0 not present not present  0
3:   35      0                 censored  level.1   2 not present not present  1
4:   99      2       death.other.causes  level.0   2 not present not present  2
5:  185      1 death.malignant.melanoma  level.2   2     present     present 12
6:  204      1 death.malignant.melanoma  level.2   2 not present     present  4
```

[†] https://github.com/amnudn/statelearner

Prototype[†]

```r
library(glmnet)
library(randomForestSRC)
lib <- list(
  cox = list(model = "cox", x_form = ~sex+age+logthick),
  cox_penalty = list(model = "GLMnet", x_form = ~invasion+ici+
     epicel+ulcer+sex+age+logthick),
  km = list(model = "cox", x_form = ~1),
  cox_strat = list(model = "cox", x_form = ~strata(epicel)),
  rf = list(model = "rfsrc", x_form = ~invasion+ici+epicel+ulcer+
     sex+age+logthick, ntree = 50)
)
```

[†] https://github.com/amnudn/statelearner

Prototype[†]

```r
set.seed(111)
sl = statelearner(learners = list(cause1 = lib,
                  cause2 = lib,
                  censor = lib),
          data = Melanoma,
          time = 5*365.25)
```

```r
sl$cv_fit
```

```
      cause1      cause2      censor     loss b
  1:      rf          km         cox 239.6142 1
  2:      rf          km cox_penalty 239.8218 1
  3:      rf          km          km 239.8678 1
  4:      rf cox_penalty         cox 239.9478 1
  5:      rf          km          rf 239.9732 1
 ---
121: cox_strat       cox          km 258.8383 1
122:      km         cox cox_penalty 259.0642 1
123: cox_strat       cox   cox_strat 259.0704 1
124:      km         cox          km 259.1725 1
125:      km         cox   cox_strat 259.3449 1
```

[†] https://github.com/amnudn/statelearner

Some theoretical results

† van der Laan and Dudoit [2003], van der Vaart et al. [2006]

Some theoretical results

Strictly proper scoring rule

We have that

$$F_P = \underset{F}{\operatorname{argmin}}\, P[\bar{B}_\tau(F, \cdot)],$$

for

$$F_P(t, j, x) = P(\eta(t) = j \mid X = x).$$

† van der Laan and Dudoit [2003], van der Vaart et al. [2006]

Some theoretical results

Strictly proper scoring rule

We have that

$$F_P = \operatorname*{argmin}_F P[\bar{B}_\tau(F, \cdot)],$$

for

$$F_P(t, j, x) = P(\eta(t) = j \mid X = x).$$

Finite sample guarantee†

For all $\delta > 0$ and $n \in \mathbb{N}$,

$$\mathbb{E}_P\left[\|\hat{\varphi}_n(\mathcal{D}_n^{-k}) - F_P\|_P^2\right] \leq (1 + 2\delta)\mathbb{E}_P\left[\|\tilde{\varphi}_n(\mathcal{D}_n^{-k}) - F_P\|_P^2\right] + (1 + \delta)16K\tau\left(13 + \frac{12}{\delta}\right)\frac{\log(1 + |\mathcal{F}_n|)}{n}.$$

† van der Laan and Dudoit [2003], van der Vaart et al. [2006]

Proof of concept

Wrongly assuming (completely) independent censoring can lead to poor performance of IPCW based super learner.

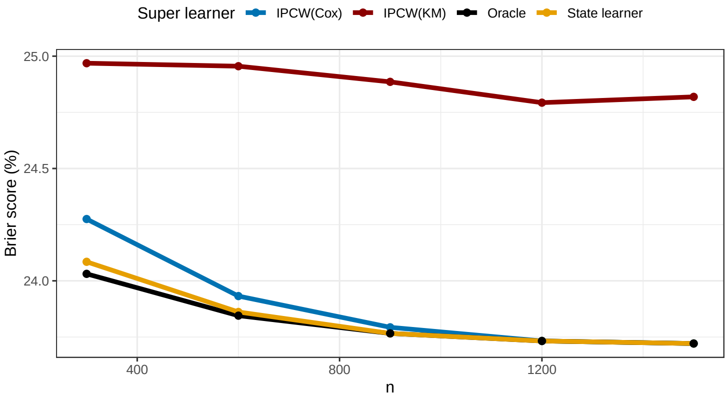
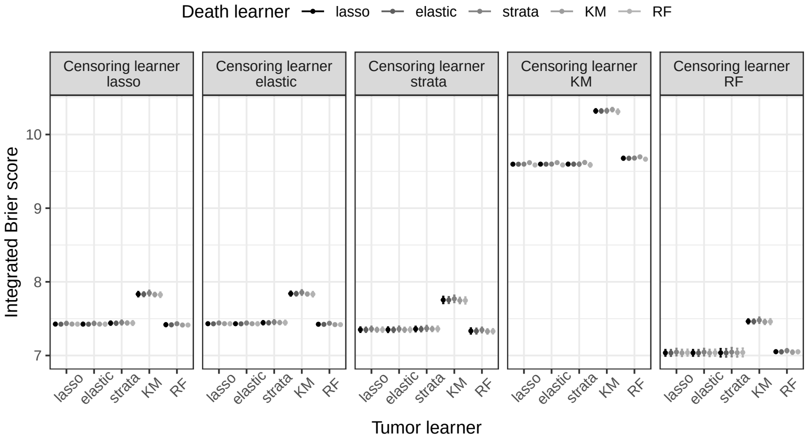


Illustration on real data with competing risks[†]



[†] Data from a prostate cancer studied by Kattan et al. [2000].

Use with targeted learning

$$F \longleftrightarrow (\Lambda, \Gamma)$$

$$\Lambda(t \mid x) = \int_0^t \frac{F(\mathrm{d}s, 1, x)}{F(s-, 0, x)}, \quad \text{and} \quad \Gamma(t \mid x) = \int_0^t \frac{F(\mathrm{d}s, 2, x)}{F(s-, 0, x)}.$$

The efficient influence function for $\tilde{\Psi}_t$ can be indexed by (Λ, Γ) or F.

The output from the state learner can be applied to construct a targeted estimator.

Second order remainder term

Important property that

$$\mathrm{Rem}(\hat{P}_n, P) = \Psi(P) - \Psi(\hat{P}_n) - P[\psi(\cdot, \hat{P}_n)]$$

is of second order.

For survival problems, $\mathrm{Rem}(\hat{P}_n, P)$ is typically dominated by terms of the form

$$\mathbb{E}_P\left[\int_0^t \hat{w}(s, X)\hat{M}_1(s \mid X)\hat{M}_2(\mathrm{d}s \mid X)]\right],$$

where $\hat{M}_j$ is either $[\Lambda - \hat{\Lambda}_n]$ or $[\Gamma - \hat{\Gamma}_n]$.

Second order remainder with the state learner

Second order property remains, in the sense that the remainder is dominated by terms of the form,

$$\mathbb{E}_P\left[\int_0^t \hat{w}_*(s, X)[F - \hat{F}](s-, j, X)[F - \hat{F}](\mathrm{d}s, j', X)\right],$$

Second order in terms of convergence rate.

Ongoing work

Extension to continuous super learner. How to best construct a convex combination?

$$\sum_{(a,b)} \alpha_{a,b}\varphi_{a,b}, \quad \text{or} \quad \varphi_{\sum_a w_a a, \sum_b w_b b},$$

where

$$\varphi_{a,b}(\mathcal{D}_n)(t,1,x) = \int_0^t e^{-a(\mathcal{D}_n)(s|x)-b(\mathcal{D}_n)(s|x)} a(\mathcal{D}_n)(\mathrm{d}s \mid x),$$

$$\dots$$

Simulation study to assess effect on low-dimensional target parameters.

Better implementation. Make more learners available.

Discussion and open questions

Limitation that F is a feature of the observed data distribution. Important whether we estimate F or (Λ, Γ) when the parameter of interest is $\Psi \colon \mathcal{P} \to \mathbb{R}$?

Second order remainder in terms of rates. Some double robustness property might be lost. Relying too much on good estimation of all of F?

Can we build or good risk prediction from the state learner?

Will a targeted estimator based on the state learner be robust against or sensitive to near positivity violations?

Summary

- Aim to avoid the need to pre-specify a censoring model.
- Select a tuple (or triple) of learners (Λ, Γ) jointly optimal for predicting the states occupied by the observed data
- No need to estimate additional nuisance parameters in the hold-out sample.
- No need to model Lebesgue densities or hazards.
- Drawback that the state learner is tuned for the a feature of the observed data distribution.

References

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

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

T. A. Gerds and M. Schumacher. Consistent estimation of the expected Brier score in general survival models with right-censored event times. *Biometrical Journal*, 48(6):1029–1040, 2006.

P. Gonzalez Ginestet, A. Kotalik, D. M. Vock, J. Wolfson, and E. E. Gabriel. Stacked inverse probability of censoring weighted bagging: A case study in the infcarehiv register. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 70(1):51–65, 2021.

E. Graf, C. Schmoor, W. Sauerbrei, and M. Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, 18(17-18):2529–2545, 1999.

X. Han, M. Goldstein, A. Puli, T. Wies, A. Perotte, and R. Ranganath. Inverse-weighted survival games. *Advances in Neural Information Processing Systems*, 34, 2021.

T. Hothorn, P. Bühlmann, S. Dudoit, A. Molinaro, and M. J. van der Laan. Survival ensembles. *Biostatistics*, 7(3): 355–373, 2006.

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*, 18(19):3352–3359, 2000.

E. C. Polley and M. J. van der Laan. Super learning for right-censored data. In M. J. van der Laan and S. Rose, editors, *Targeted Learning: Causal Inference for Observational and Experimental Data*, pages 249–258. Springer, 2011.

M. C. Sachs, A. Discacciati, Å. H. Everhov, O. Olén, and E. E. Gabriel. Ensemble prediction of time-to-event outcomes with competing risks: A case-study of surgical complications in Crohn's disease. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 68(5):1431–1446, 2019.

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. Dudoit. Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples. Technical report, Division of Biostatistics, University of California, 2003.

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

A. W. van der Vaart, S. Dudoit, and M. J. van der Laan. Oracle inequalities for multi-fold cross validation. *Statistics & Decisions*, 24(3):351–371, 2006.

T. Westling, A. Luedtke, P. Gilbert, and M. Carone. Inference for treatment-specific survival curves using machine learning. *arXiv preprint arXiv:2106.06602*, 2021.

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