The state learner

Anders Munch
joint work with Thomas Gerds

October 10, 2023

Outline

Super learning (aka cross-validation, stacked regression, ...[1])

Example: Consider estimating a conditional mean $f(x) = \mathbb{E}[Y \mid X = x]$ based on data $\mathcal{D}_n = \{O_1, \dots, O_n\}$, where $O_i = (X_i, Y_i)$ are iid. observations.

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 $L(a(\mathcal{D}_n), O)$, e.g., $L(a(\mathcal{D}_n), O) = \{a(\mathcal{D}_n)(X) - Y\}^2$

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

Super learning (aka cross-validation, stacked regression, ...[1])

Example: Consider estimating a conditional mean $f(x) = \mathbb{E}[Y \mid X = x]$ based on data $\mathcal{D}_n = \{O_1, \dots, O_n\}$, where $O_i = (X_i, Y_i)$ are iid. observations.

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 $L(a(\mathcal{D}_n), O)$, e.g., $L(a(\mathcal{D}_n), O) = \{a(\mathcal{D}_n)(X) - Y\}^2$

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

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

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

Super learning (aka cross-validation, stacked regression, ...[1])

Example: Consider estimating a conditional mean $f(x) = \mathbb{E}[Y \mid X = x]$ based on data $\mathcal{D}_n = \{O_1, \ldots, O_n\}$, where $O_i = (X_i, Y_i)$ are iid. observations.

- **Learner** algorithm a that produces estimates, $\mathcal{D}_n \mapsto a(\mathcal{D}_n) = \hat{f}_n$
- **Library** collection of learners, $\mathcal{A} = \{a_1, a_2, \ldots, a_M\}$
- **Loss function** $L(a(\mathcal{D}_n), O)$, e.g., $L(a(\mathcal{D}_n), O) = \{a(\mathcal{D}_n)(X) - Y\}^2$
- **Discrete SL** $\hat{a}_n = \text{argmin}_{a \in \mathcal{A}} \hat{R}_n(a; L)$, where

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

A super learner can be used for

- model selection and hyperparameter tuning
- stand-alone prediction
- nuisance parameter estimation (e.g., targeted learning of ATE)

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

Right-censored data

Notation

X	vector of baseline covariates
T	time to event variable, $T > 0$
C	censoring time, $C > 0$
$\tilde{T}$	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, $O = (X, \tilde{T}, \Delta) \sim P$
Q	distribution of the data of interest $(X, T) \sim Q$

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

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

We assume $T \perp\!\!\!\perp C \mid X$, which implies that Λ and Γ are identifiable from P.

Super learning with right-censored data

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

Q distribution of the data of interest $(X, T) \sim Q$

In a survival context, we have data $\mathcal{D}_n = \{O_1, \dots, O_n\}$, from P, but we are interested in (a feature of) Q, such as Λ.

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

Super learning with right-censored data

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

Q distribution of the data of interest $(X, T) \sim Q$

In a survival context, we have data $\mathcal{D}_n = \{O_1, \dots, O_n\}$, from P, but we are interested in (a feature of) Q, such as Λ.

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

The challenge of censoring

$a(\mathcal{D}_n^{-k})$ Many learners are available for this type of data (e.g., semi-parametric Cox models, parametric survival models, (stratified) Kaplan-Meier estimators, random survival forest) ✓

$L(a(\mathcal{D}_n^{-k}), O_i)$ How to evaluate the performance of a learner trained in $\mathcal{D}_n^{-k}$ in the hold-out data $\mathcal{D}_n^{-k}$?

Existing approaches

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

Requires discrete time or modeling a Lebesgue hazard function which is incompatible with many common estimators in survival analysis (e.g., Kaplan-Meier, semi-parametric Cox models, and random survival forests).

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

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

IPCW (e.g., Hothorn et al. [2006], Gonzalez Ginestet et al. [2021])

Inverse probability of censoring weighted loss functions also require a pre-specified censoring model.

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

To avoid this, it has been suggested to iterate between estimation of Λ and Γ. No theoretical guarantees for this procedure.

The observed multi-state system

Modeling the conditional state-occupation probabilities of the *observed* data.

The observed multi-state system

Modeling the conditional state-occupation probabilities of the *observed* data.

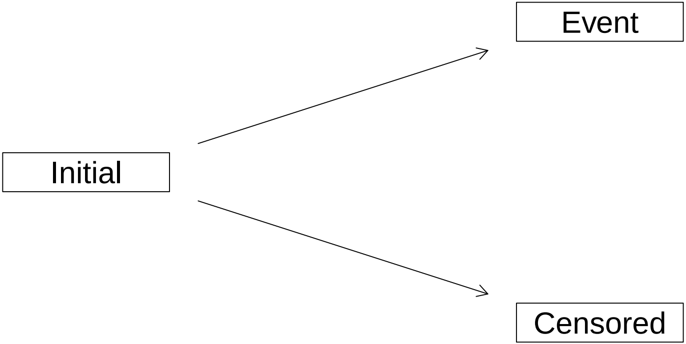
$$(X, T) \sim Q$$

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

The observed multi-state system

Modeling the conditional state-occupation probabilities of the *observed* data.

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



Conditional state-occupation probabilities for observed data

Record the observed data as $O = (X, \{\eta(t) : t \geq 0\})$, where

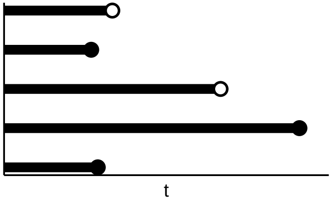
$$\eta(t) = \mathbb{1}\{\tilde{T} \leq t, \Delta = 1\} + 2\,\mathbb{1}\{\tilde{T} \leq t, \Delta = 0\} \in \{0, 1, 2\}.$$

Denote by

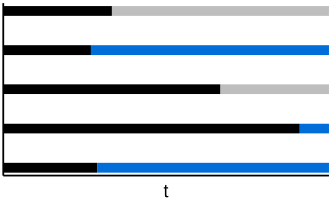
$$F(t, j, x) = P(\eta(t) = j \mid X = x), \quad \text{for all } t \geq 0,\ j \in \{0, 1, 2\},\ x \in \mathbb{R}^d,$$

the conditional state-occupation probabilities for the observed data.

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



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



The state learner

The state learner builds a super learner for the conditional state-occupation probabilities,

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

F is a feature of the observed data distribution P, so performance can be evaluated directly as in a "non-survival" setting.

We suggest to use the integrated Brier score $\bar{B}_\tau(F, O) = \int_0^\tau B_t(F, O)\,\mathrm{d}t$, where

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

With this choice of loss function no modeling of Lebesgue hazards or densities is required.

Expressing F using Λ and Γ

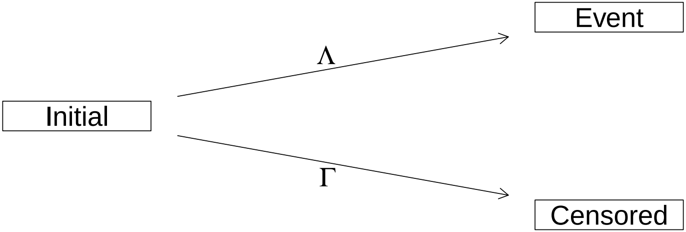
$$F(t, j, x) = P(\eta(t) = j \mid X = x), \quad \text{for all } t \geq 0,\ j \in \{0, 1, 2\},\ x \in \mathbb{R}^d$$

can be expressed (slightly informally) using Λ and Γ,

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

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

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



Constructing a library for learning F

Many learners for Λ (and Γ) are avalaible (Cox models, random survival forests, etc.).

Given libraries $\mathcal{A}$ and $\mathcal{B}$ for learning Λ and Γ, respectively, we construct the library

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

where

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

$$\cdots$$

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

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

Some theoretical results

Finite sample guarantee

Using results from [van der Laan and Dudoit, 2003, van der Vaart et al., 2006] we can establish a finite sample oracle inequality for the state learner.

This means that the state learner will perform almost as well as a so-called "oracle" which uses the unknown data-generating distribution to evaluate performance of the learners.

Asymptotic consequence

Let F_0 denote the conditional state-occupation probability function corresponding to the underlying data-generating distribution P_0. If

- $|\mathcal{F}(\mathcal{A}_n, \mathcal{B}_n)| = O(n^q)$, for some $q \in \mathbb{N}$, and
- the library contains a learner that converges to F_0 at rate r_n,

then the state learner converges to F_0 at the same rate or at rate $\log(n) r_n$.

Almost minimum viable product

```r
head(use_dat, n=4)
```

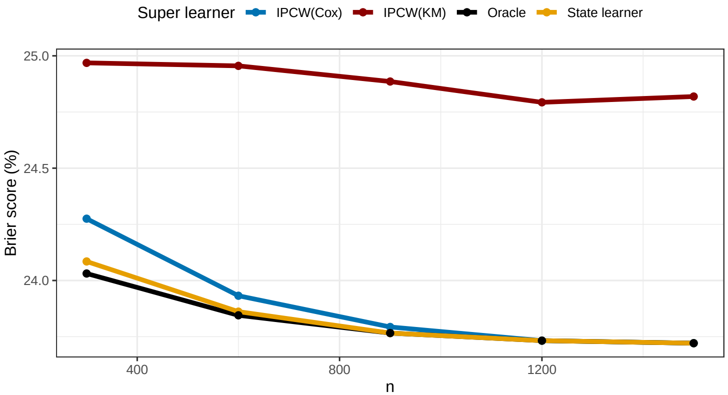
```
        time status   logPSA stage ggtot      sDose hormones
1: 30.78737      0 1.791759   T1c     6  0.1663670       No
2: 28.69895      0 2.468100   T3c     9  0.1663670      Yes
3: 11.99158      0 3.086487   T1c     3 -0.9372808       No
4: 38.13053      1 2.890372   T1c     6 -0.9372808       No
```

```r
library <- list(
  cox_lasso = list("GLMnet"),
  cox_elastic = list("GLMnet", alpha = 0.5),
  rf = list("rfsrc", ntree = 500))
fit_sl <- statelearner(
  list(cause1 = library, censor = library),
  data = use_dat, time = 36),
head(fit_sl, n=4)
```

```
        cause1 censor     loss         sd
1: cox_elastic     rf 7.034702 0.02159417
2: cox_elastic     rf 7.034812 0.02286074
3:   cox_lasso     rf 7.035051 0.02142064
4:   cox_lasso     rf 7.035231 0.02266556
```

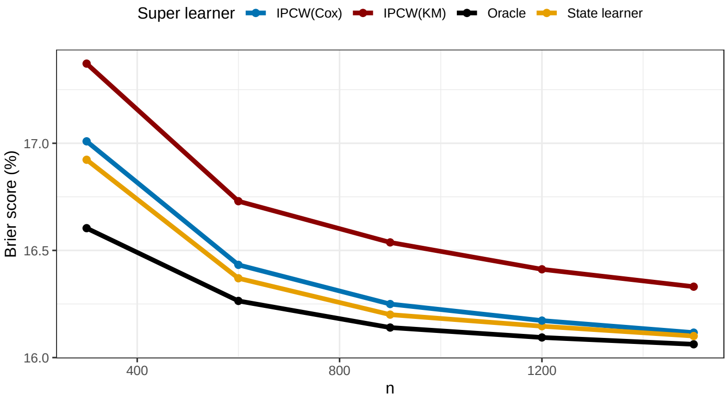
Proof of concept – simulation I

- Univariate X
- Cox model and the Nelson-Aalen estimator in the libraries
- Compare to IPCW weighted estimators using wrongly (IPCW(KM)) and correctly (IPCW(Cox)) specified censoring models
- Evaluate performance of survival predictions at fixed prediction horizon

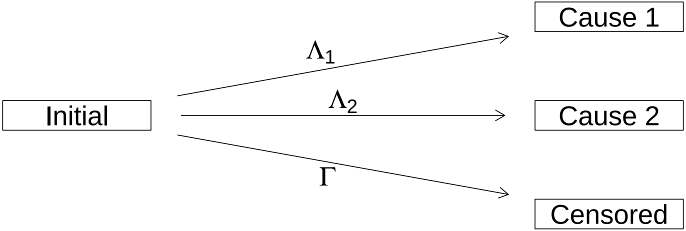


Proof of concept – simulation II

- Multivariate X
- Several strong learners: Cox models (various stratifications and splines), penalized Cox models (lasso, ridge, elastic), random survival forest
- Data generated according to a simulation of a prostate cancer study [Kattan et al., 2000, Gerds et al., 2013].



Competing risks



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

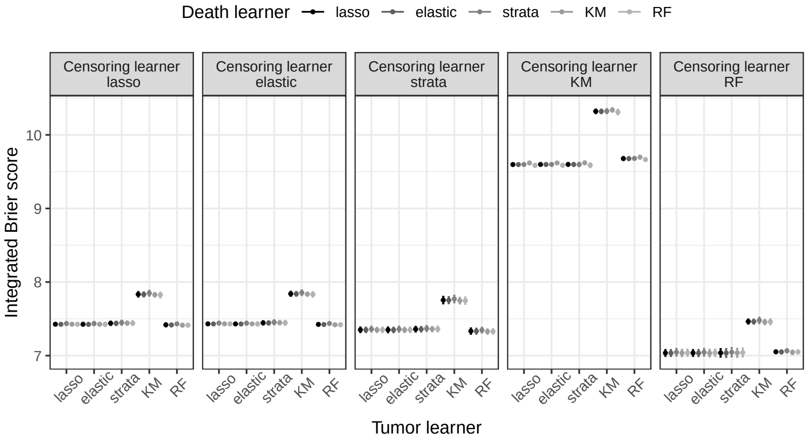
$$F(t, 1, x) = P(\tilde{T} \leq t, \tilde{D} = 1 \mid X = x) = \int_0^t e^{-\Lambda_1(s|x) - \Lambda_2(s|x) - \Gamma(s|x)} \Lambda_1(\mathrm{d}s \mid x),$$

$$\ldots$$

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

Proof of concept – some real data

The real data considered in [Kattan et al., 2000] included the competing risk of death.



Discussion

A clear limitation is that the function F is typically not a parameter of interest.

We can obtain a risk prediction model from the state learner using that

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

However, the state learner does not evaluate the learners based on their risk prediction performances but on how well a tuple (Λ, Γ) of learners jointly model the observed data.

When estimating low-dimensional target parameter and the state learner is used to estimate the nuisance parameters, this is probably less of a concern.

Unclear if the state learner will respond well to positivity violations or not.

Conclusion

- To avoid the need to pre-specify a censoring model, we propose to use learners for Λ and Γ to jointly model the observed data.
- We select a tuple of learners (Λ, Γ) that is jointly optimal for predicting the states occupied by the observed data conditional on baseline covariates.
- We use the integrated Brier score to evaluate performance with respect to the observed data distribution.
- No need to model additional nuisance parameters to estimate performance in hold-out samples.
- No need to estimate a Lebesgue densities or hazards.
- Drawback is that the SL is tuned for the a feature of the observed distribution P and not for a feature of Q.

Questions, comments, suggestions?

Thank you for listening!

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, M. W. Kattan, M. Schumacher, and C. Yu. Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. *Statistics in medicine*, 32(13):2173–2184, 2013.

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.

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.