Loss functions and cross-validation with censored survival data

Anders Munch
joint work with Thomas Gerds

PhD Student, Section of Biostatistics
University of Copenhagen

September 7, 2022 – JICI

Outline

Setting and data structure

Loss functions and hold-out samples for survival data

Double robustness and fluctuation risk

Data structure and target of inference

Survival setting

$O = (\tilde{T}, \Delta, X) \sim P \in \mathcal{P}$ Oberved data with $\mathcal{O} = \mathbb{R}_+ \times \{0, 1\} \times \mathbb{R}^p$.

$Z = (T, X) \sim Q \in \mathcal{Q}$ The distribution Q (or a feature of it) is of interest.

Parameters of interest

Low-dimensional feature of Q, e.g., the marginal survival probability $Q(T > t)$ for a fixed time horizon $t \in \mathbb{R}_+$.

Nuisance parameter

The distribution Q is identifiable from the observed data distribution P under coarsening at random. Without further assumptions we would typically need to estimate the conditional survival function $S(t \mid x) = Q(T > t \mid X = x)$ (and/or the conditional censoring distribution).

Cross-validation and super learning for S

Cross-validation and super learning for S

Most machine learning methods depend on one or more hyperparameters which are typically chosen using **cross-validation**.

More generally, to build robust estimators we can use **stacked regression** / **super learning** [Breiman, 1996, van der Laan et al., 2007] to select from or combine a collection candidate estimators/algorithms.

A central component for both cross-validation and super learning is the partitioning of data into training and test folds. A suitable loss function is then used to evaluate the performance of an estimator in hold-out samples.

Evaluate performance in hold-out samples

D data set $(O_1, \ldots, O_n)$

$\mathcal{A}$ collection of algorithms for estimating $S \in \mathcal{S}$

$\nu \in \mathcal{A}$ mapping $D \longmapsto \nu(D) = \hat{S} \in \mathcal{S}$

L loss function, $L \colon \mathcal{S} \times \mathcal{O} \to \mathbb{R}_+$

To evaluate the performance of $\nu \in \mathcal{A}$ let

$D_1, \ldots, D_K$ partition of the data set D

D_{-k} the k'th training sample, $D_{-k} = D \setminus D_k$, $k = 1, \ldots, K$

and evaluate for all $i = 1, \ldots, n$,

$$L(\nu(D_{-k}), O_i), \quad \text{for all} \quad O_i \in D_k.$$

Averaging these values gives us an estimate of the expected loss of the algorithm $\nu \in \mathcal{A}$, and we can then pick the one with lowest expected loss. Alternatively, we can use these value to combine all algorithms into a super learner.

The partial likelihood and hold-out samples

A popular choice for training survival models is the negative partial log-likelihood loss. Assuming conditional independence between the outcome and the censoring given the covariates, the observed data factorizes as

$$\ell(P, O) = \ell_t(S, O) \cdot \ell_c(G, O) \cdot \ell_0(h, O),$$

where $G \in \mathcal{G}$ denotes the censoring mechanism and h the marginal distribution of the baseline covariates. The negative partial log-likelihood for the component S is

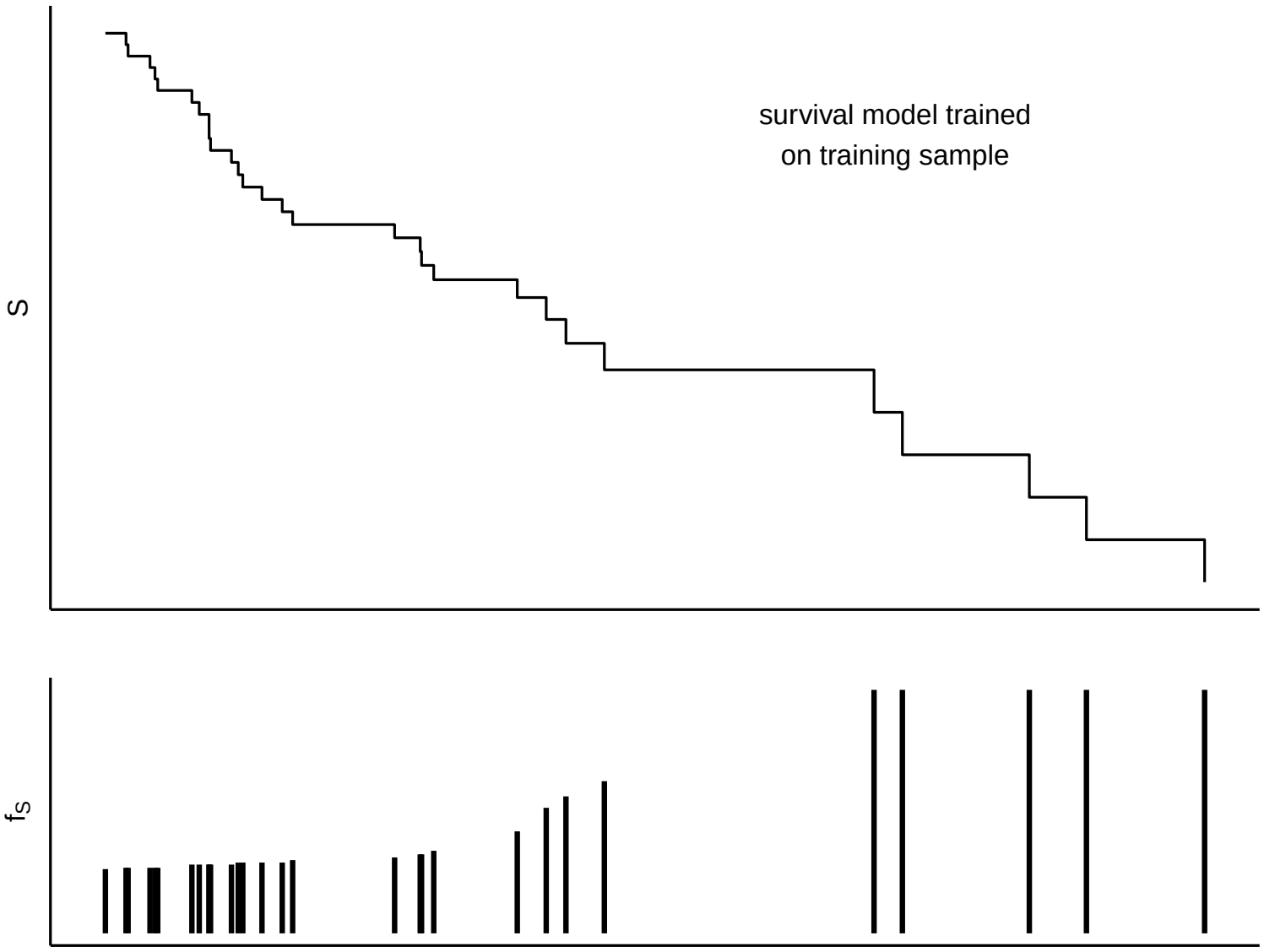
$$-\log \ell_t(S, O) = -\left\{(1 - \Delta) \log S(\tilde{T} \mid X) + \Delta \log f_S(\tilde{T} \mid X)\right\},$$

where f_S is the conditional density corresponding to S.

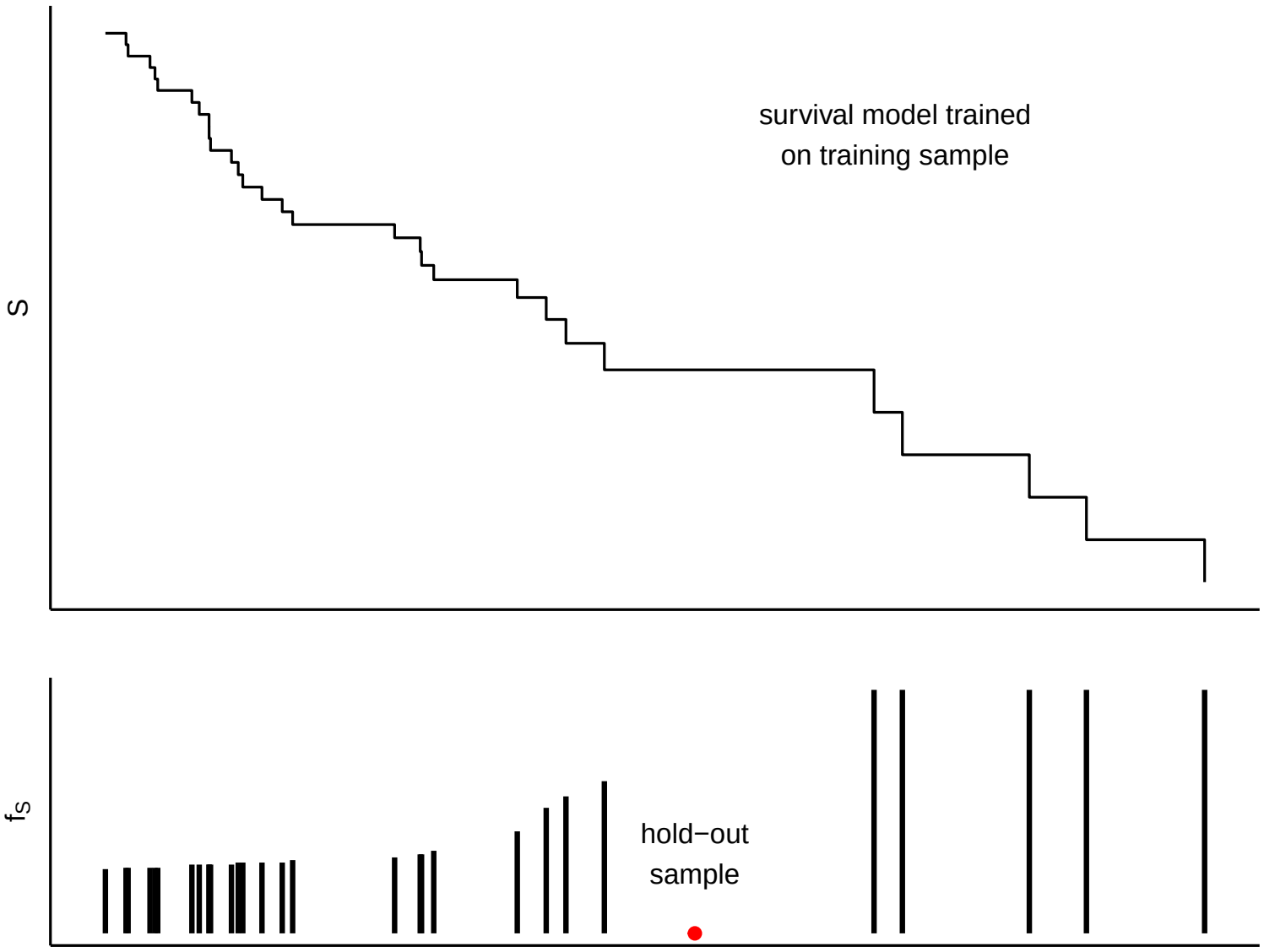
However, in continuous time this loss function is unsuitable for evaluating performance of most common survival estimators in hold-out samples, because (a.s.)

$$f_{\hat{S}}(\tilde{T}_i \mid X_i) = 0 \quad \text{when} \quad \hat{S} = \nu(D_{-k}) \quad \text{and} \quad (\tilde{T}_i, \Delta_i, X_i) \in D_k.$$

Illustration



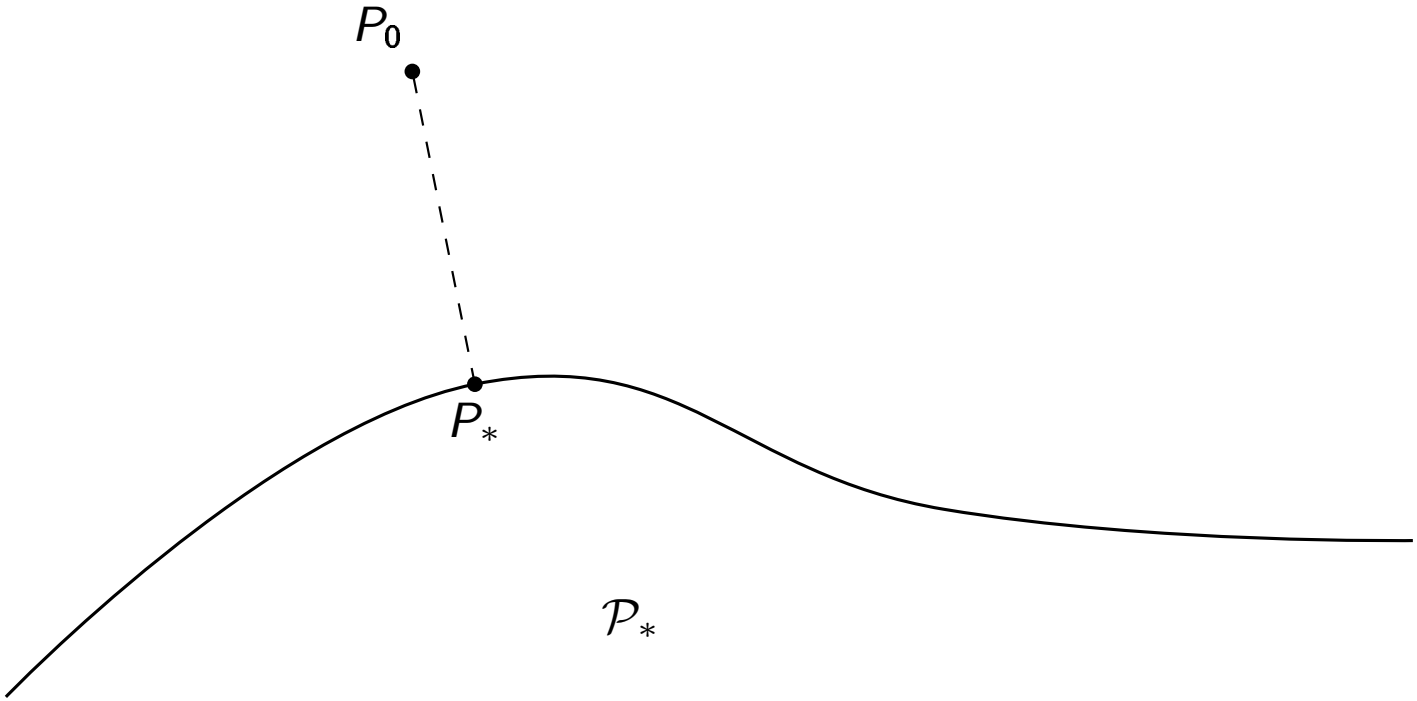
Illustration



Kullback-Leibler divergence and partial likelihoods

Maximum likelihood estimation is connected to minimizing the Kullback-Leibler divergence and gives an interpretation of the MLE under misspecified models.

$$D_{\mathrm{KL}}(P_0 \,||\, P) := P_0\left[\log \frac{p_0}{p}\right], \quad \text{where} \quad P_0 = p_0 \cdot \mu, P = p \cdot \mu.$$



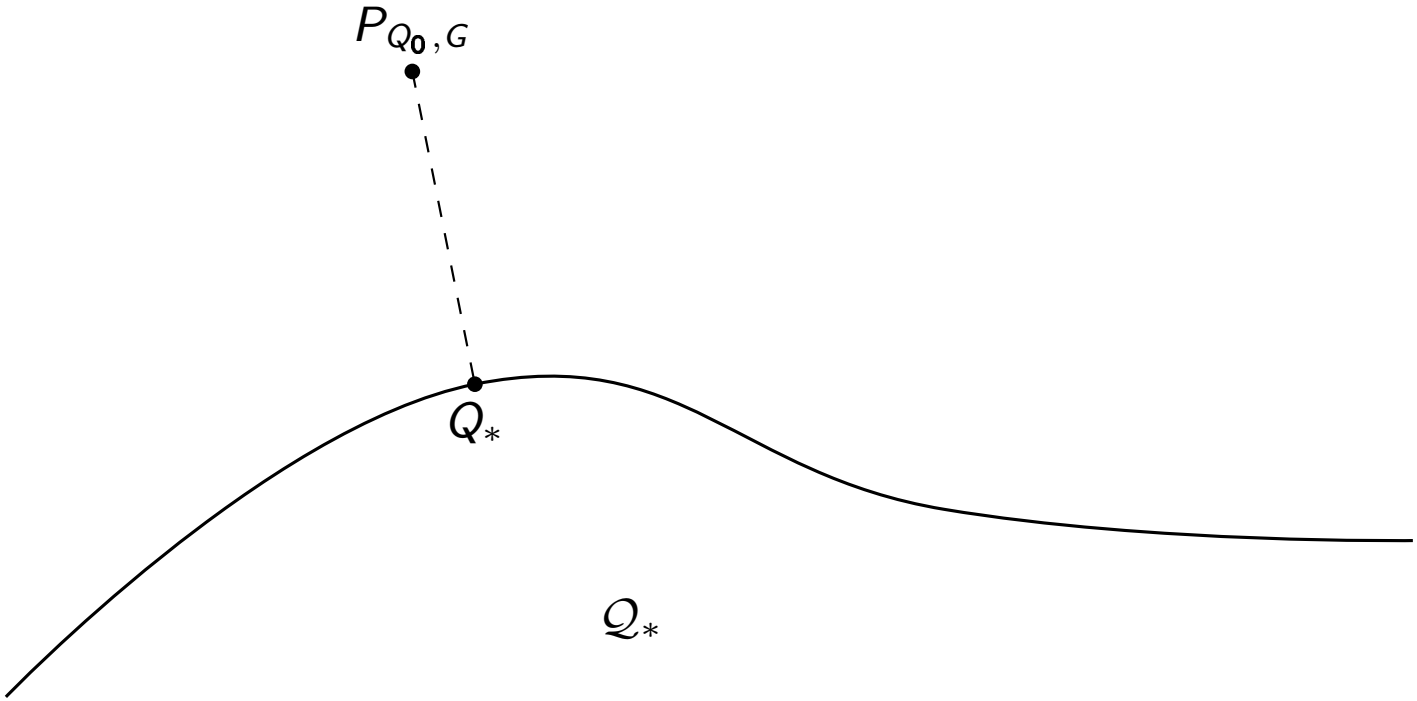
Kullback-Leibler divergence and partial likelihoods

Maximum likelihood estimation is connected to minimizing the Kullback-Leibler divergence and gives an interpretation of the MLE under misspecified models.

$$D_{\mathrm{KL}}(P_0 \,||\, P) := P_0\left[\log \frac{p_0}{p}\right], \quad \text{where} \quad P_0 = p_0 \cdot \mu, P = p \cdot \mu.$$

With partial likelihood we are minimizing

$$Q \longmapsto D_{\mathrm{KL}}(P_{Q_0,G} \,||\, P_{Q,G}), \quad \text{with} \quad Q \in \mathcal{Q}_*.$$



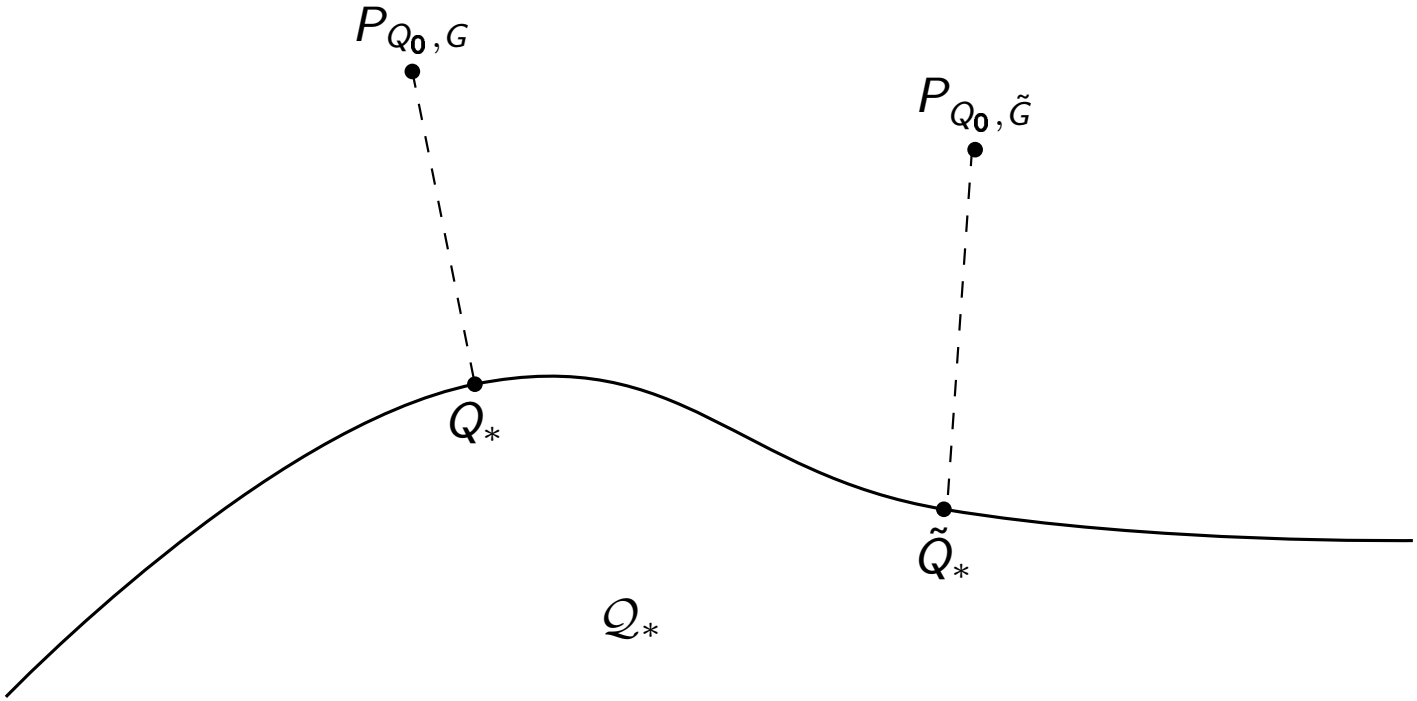
Kullback-Leibler divergence and partial likelihoods

Maximum likelihood estimation is connected to minimizing the Kullback-Leibler divergence and gives an interpretation of the MLE under misspecified models.

$$D_{\mathrm{KL}}(P_0 \,||\, P) := P_0\left[\log \frac{p_0}{p}\right], \quad \text{where} \quad P_0 = p_0 \cdot \mu, P = p \cdot \mu.$$

With partial likelihood we are minimizing

$$Q \longmapsto D_{\mathrm{KL}}(P_{Q_0,G} \,||\, P_{Q,G}), \quad \text{with} \quad Q \in \mathcal{Q}_*.$$



Inverse probability of censoring weighted loss functions

A conceptually more attractive strategy is to use loss functions that are

(i) suited for evaluating the performance of estimating the *survival function*

(ii) defined in terms of the *distribution Q of interest*

We can do this using inverse probability of censoring weighted (IPCW) loss functions. For instance, with the Brier score

$$L_{\text{Brier}}(S, Z) = (S(t \mid X) - \mathbb{1}\{T > t\})^2, \quad Z = (T, X) \sim Q,$$

the expected loss is identifiable through

$$\mathbb{E}_Q\left[L_{\text{Brier}}(S, Z)\right] = \mathbb{E}_P\left[W_G \cdot L_{\text{Brier}}(S, Z)\right],$$

with

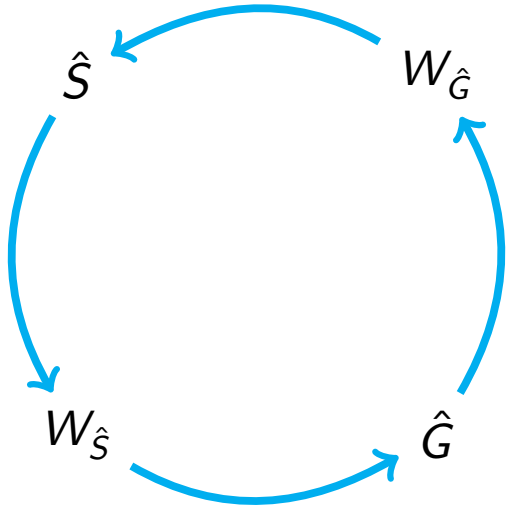
$$W_G = \frac{\mathbb{1}\{\tilde{T} > t\} + \mathbb{1}\{\tilde{T} \leq t\}\Delta}{G(\tilde{T} \wedge t \mid X)},$$

where G is the conditional "survivor" function for the censoring distribution [Graf et al., 1999, Gerds and Schumacher, 2006, van der Laan and Dudoit, 2003].

Estimation of the IPC weights

To use IPCW loss functions in practice we need to estimate G. This is the same estimation problem as estimation of S, just with the meaning of Δ reversed.

$\Longrightarrow$ The exact same challenges face us when attacking this problem.



Recently, Han et al. [2021] and Westling et al. [2021] have suggested to iterate between estimation of $\hat{S}$ and $\hat{G}$ until convergence.

Not obvious how to select our survival model

Not obvious how to select our survival model

When S is a nuisance parameter we could aim at selecting the model based on a criteria designed for the parameter of interest.

Not obvious how to select our survival model

When S is a nuisance parameter we could aim at selecting the model based on a criteria designed for the parameter of interest.

Exploiting double robustness

Cui and Tchetgen [2020], building on ideas from Robins et al. [2007], proposed to exploit double robustness as a model selection criteria.

Fluctuation risk

Let ψ be the efficient influence for the parameter Ψ, and assume we can write $\psi(O, P) = \varphi(O, S_P, G_P) - \Psi(P)$ such that

$$\mathbb{E}_P\left[\varphi(O, S_P, G_*)\right] = \mathbb{E}_P\left[\varphi(O, S_*, G_P)\right] = \Psi(P),$$

for any S_* and G_*, where S_P and G_P are the conditional survivor functions corresponding to the data generating distribution.

Fluctuation risk

Let ψ be the efficient influence for the parameter Ψ, and assume we can write $\psi(O, P) = \varphi(O, S_P, G_P) - \Psi(P)$ such that

$$\mathbb{E}_P\left[\varphi(O, S_P, G_*)\right] = \mathbb{E}_P\left[\varphi(O, S_*, G_P)\right] = \Psi(P),$$

for any S_* and G_*, where S_P and G_P are the conditional survivor functions corresponding to the data generating distribution.

Let $\mathcal{G}$ be a (finite) collection of models for G. The double robustness property implies that $\mathbb{E}_P\left[\varphi(O, S_P, G)\right] = \mathbb{E}_P\left[\varphi(O, S_P, G')\right]$ for any $G, G' \in \mathcal{G}$. In particular,

$$\max_{G, G' \in \mathcal{G}} \left| \mathbb{E}_P\left[\varphi(O, S_P, G)\right] - \mathbb{E}_P\left[\varphi(O, S_P, G')\right] \right| = 0.$$

Fluctuation risk

Let ψ be the efficient influence for the parameter Ψ, and assume we can write $\psi(O, P) = \varphi(O, S_P, G_P) - \Psi(P)$ such that

$$\mathbb{E}_P\left[\varphi(O, S_P, G_*)\right] = \mathbb{E}_P\left[\varphi(O, S_*, G_P)\right] = \Psi(P),$$

for any S_* and G_*, where S_P and G_P are the conditional survivor functions corresponding to the data generating distribution.

Let $\mathcal{G}$ be a (finite) collection of models for G. The double robustness property implies that $\mathbb{E}_P\left[\varphi(O, S_P, G)\right] = \mathbb{E}_P\left[\varphi(O, S_P, G')\right]$ for any $G, G' \in \mathcal{G}$. In particular,

$$\max_{G, G' \in \mathcal{G}} \left| \mathbb{E}_P\left[\varphi(O, S_P, G)\right] - \mathbb{E}_P\left[\varphi(O, S_P, G')\right] \right| = 0.$$

This motivates the "fluctuation risk",[1]

$$R(S) = \max_{G, G' \in \mathcal{G}} \left| \mathbb{E}_P\left[\varphi(O, S, G)\right] - \mathbb{E}_P\left[\varphi(O, S, G')\right] \right|.$$

[1] or pseudo-risk because it depends $\mathcal{G}$ which is suppressed in the notation.

Estimating the fluctuation risk

Let $\mathcal{A}_c$ be a collection of algorithms for estimating G. For any $\nu \in \mathcal{A}$, $\gamma \in \mathcal{A}_c$, and $k = 1, \ldots, K$ define

$$\hat{\Psi}^k_{\nu,\gamma} = \frac{1}{|D_k|} \sum_{O \in D_k} \varphi(O, \nu(D_{-k}), \gamma(D_{-k})).$$

For any $\nu \in \mathcal{A}$ we approximate the fluctuation risk with

$$\hat{R}(\nu) = \frac{1}{K} \sum_{k=1}^{K} \max_{\gamma,\gamma' \in \mathcal{A}_c} \left| \hat{\Psi}^k_{\nu,\gamma} - \hat{\Psi}^k_{\nu,\gamma'} \right|.$$

Recall

$\mathcal{A}$	collection of algorithms for estimating $S \in \mathcal{S}$
$D_1, \ldots, D_K$	partition of the data set D
D_{-k}	the k'th training sample, $D_{-k} = D \setminus D_k$, $k = 1, \ldots, K$

Illustration of the method

Consider the following simple setting where $X = (A_1, A_2, A_3)^T$ with $A_j \in \{0, 1\}$ for all j and our parameter of interest is the marginal survival probability $Q(T > t)$ at some fixed time $t > 0$. We consider using Kaplan-Meier estimators stratified on each of A_j.[2]

outcome algorithms

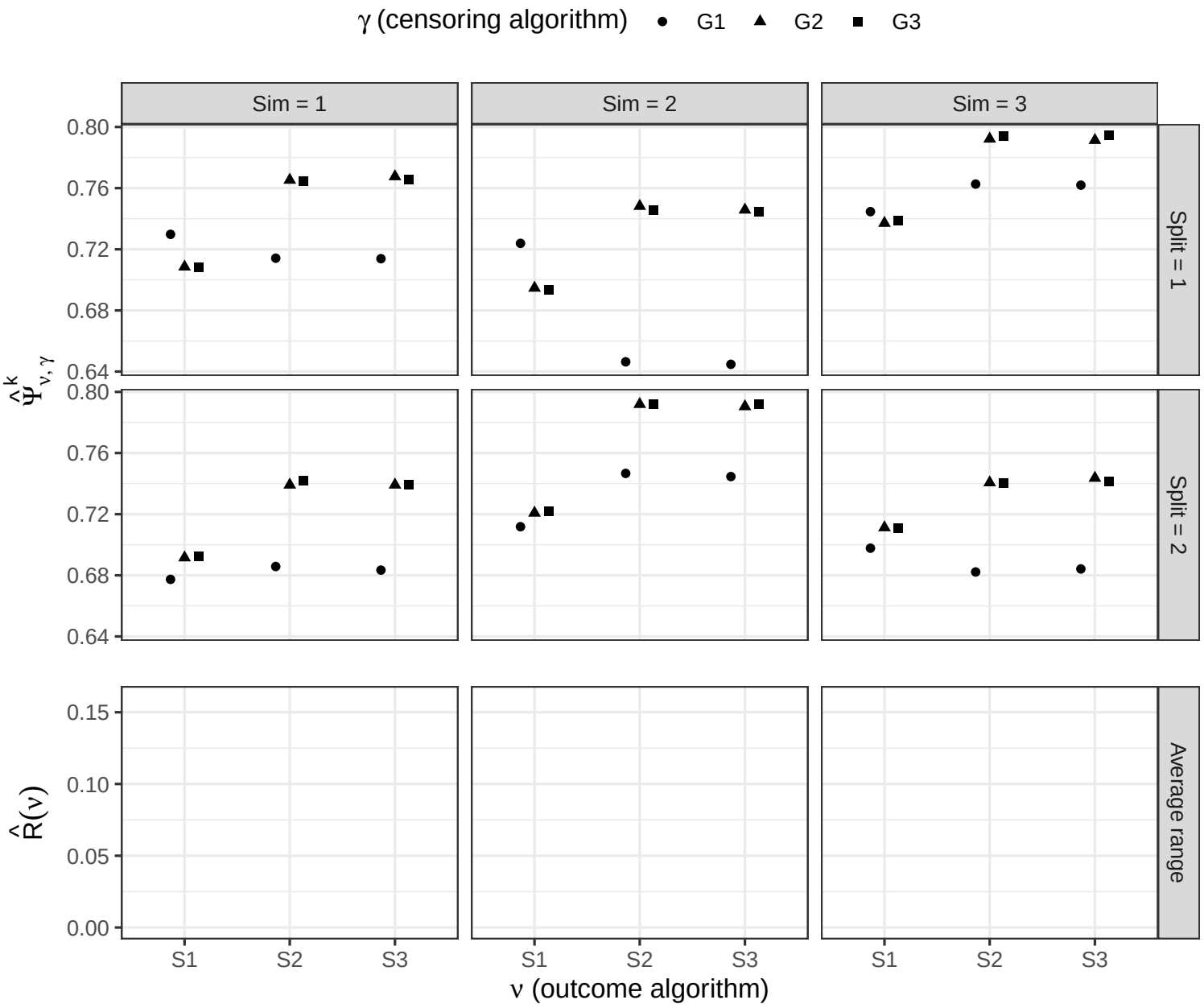
```
S1 <- function(d) prodlim(Surv(time,event) ~ A1, data = d)
S2 <- function(d) prodlim(Surv(time,event) ~ A2, data = d)
S3 <- function(d) prodlim(Surv(time,event) ~ A3, data = d)
```

censoring algorithms

```
G1 <- function(d) prodlim(Surv(time,event) ~ A1, rev = T, data = d)
G2 <- function(d) prodlim(Surv(time,event) ~ A2, rev = T, data = d)
G3 <- function(d) prodlim(Surv(time,event) ~ A3, rev = T, data = d)
```

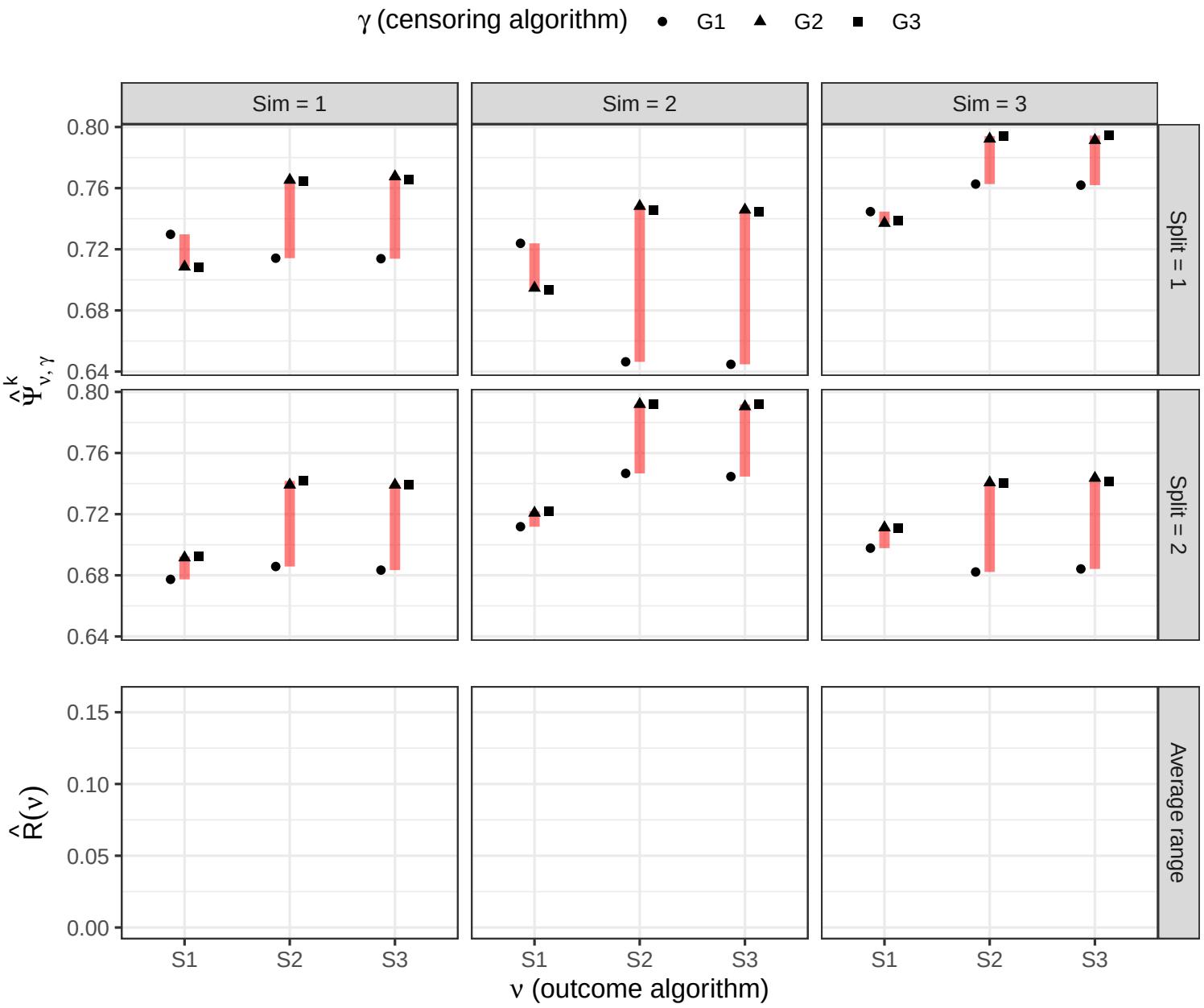
[2] In this simulation, only A_1 influences survival and censoring.

Illustration of the method



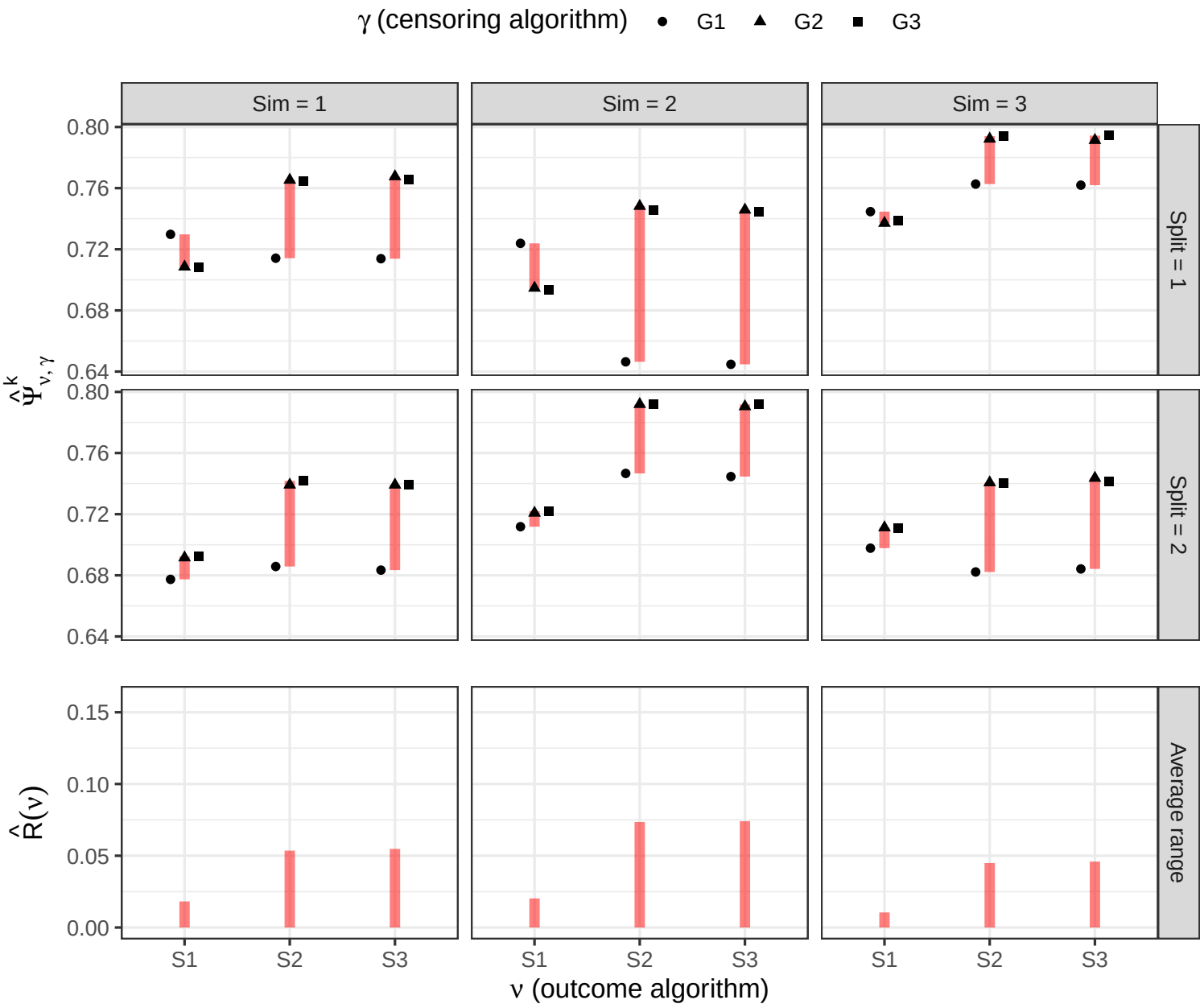
[2] In this simulation, only A_ν influences survival and censoring

Illustration of the method



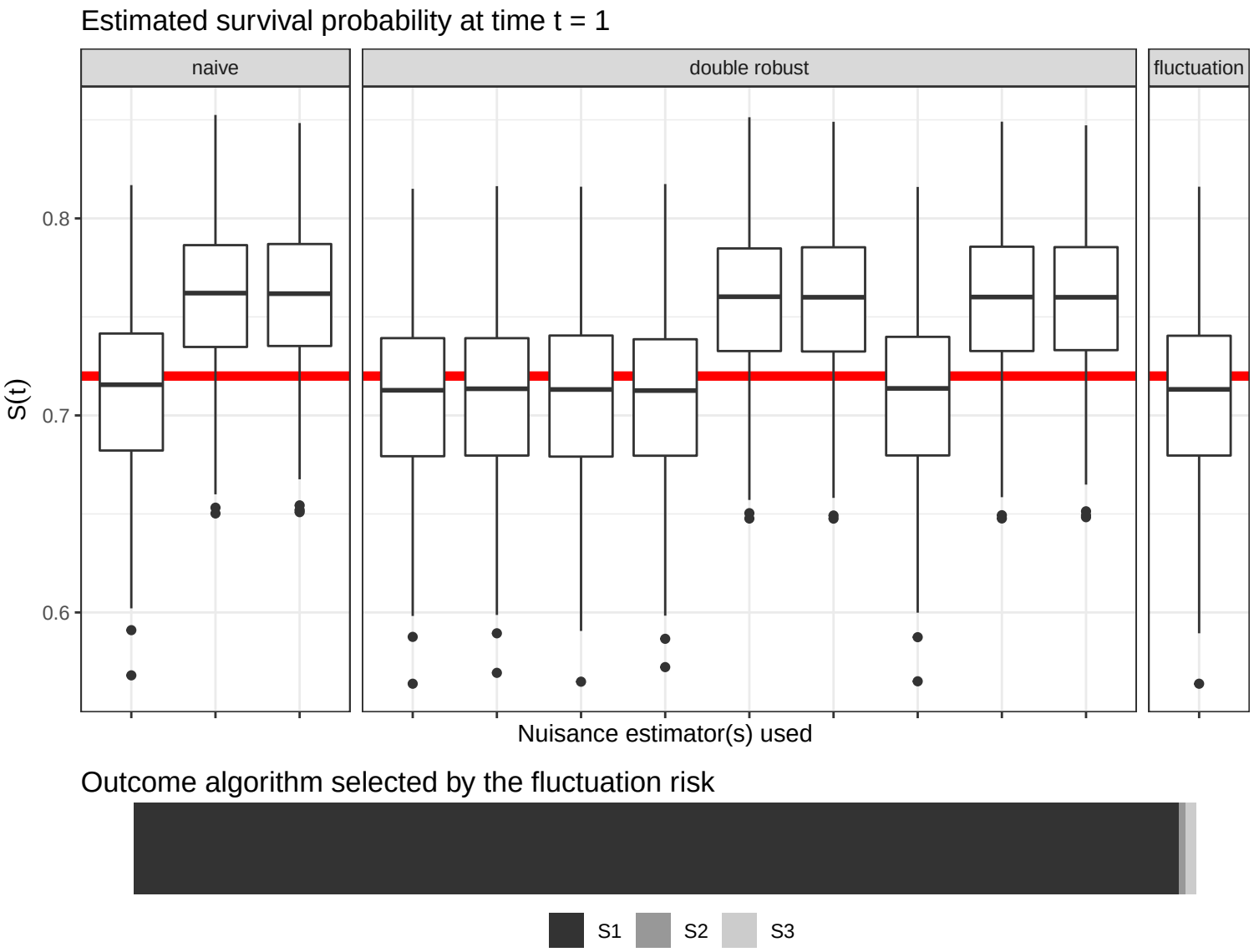
[2] In this simulation, only A_ν influences survival and censoring

Illustration of the method

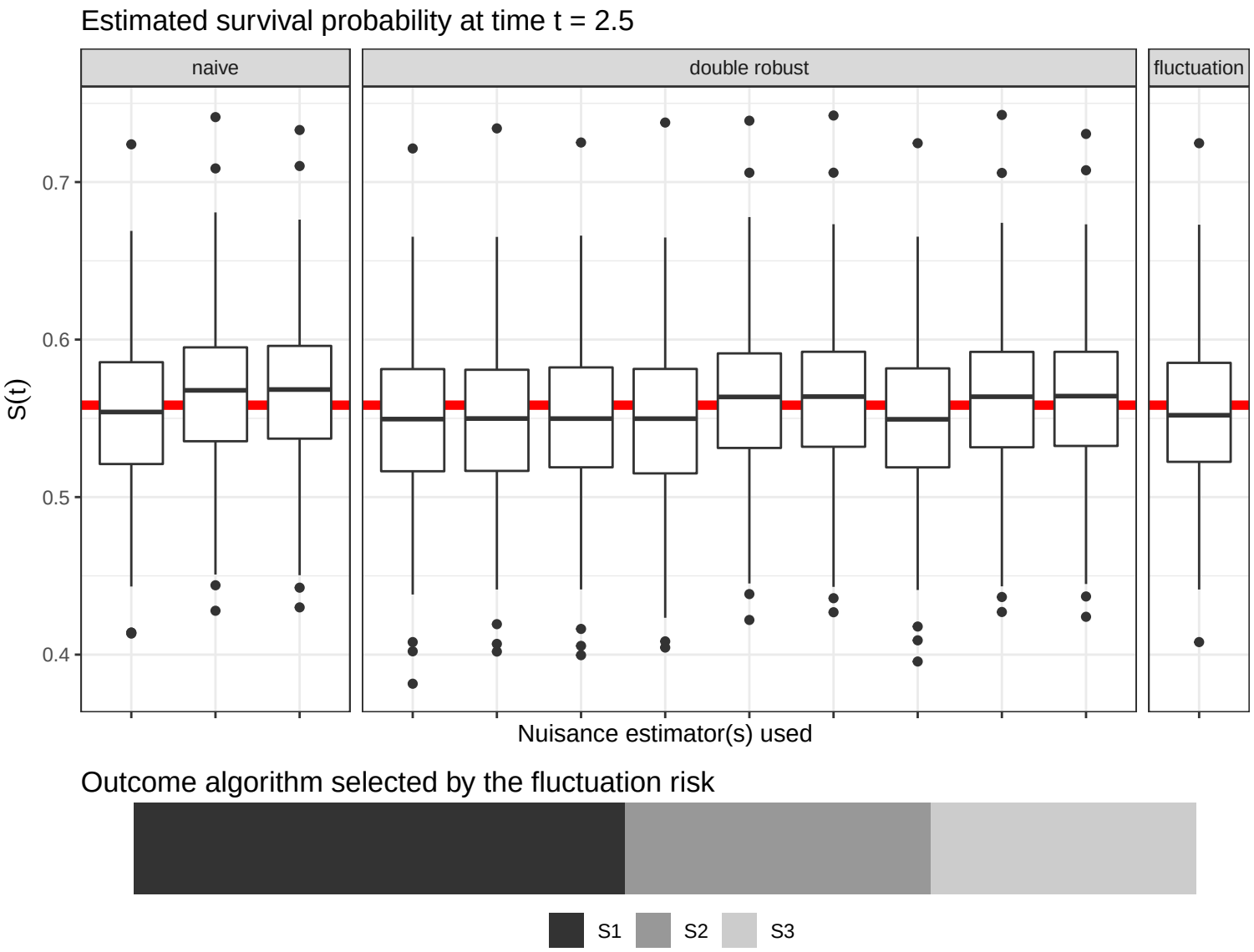


[2] In this simulation, only A_ν influences survival and censoring

Some simulation results



Some simulation results



Conclusion

- It is not obvious what loss function to use for estimating the conditional survivor function with censored data observed in continuous time.
- If the parameter of interest is a low-dimension feature of the full data distribution we could exploit this and evaluate the performance of the nuisance parameter estimators in terms of their effect on the estimator of the target parameter.

References

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

Y. Cui and E. J. Tchetgen. Selective machine learning of doubly robust functionals. *arXiv preprint arXiv:2004.03036*, 2020.

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.

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.

J. Robins, M. Sued, Q. Lei-Gomez, and A. Rotnitzky. Comment: Performance of double-robust estimators when" inverse probability" weights are highly variable. *Statistical Science*, 22(4):544–559, 2007.

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. 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.

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

If the conditional survivor function was the target parameter

Consider now the situation where the conditional survival function $S(t \mid x)$ is the actual parameter of interest for fixed t. Assume that our goal is to build a prediction model minimizing the average Brier score. Given a model S we can consider the average Brier score of S as a low dimensional target parameter

$$\Psi_S(P) = \mathbb{E}_P \left[W_G \cdot L_{Brier}(S, Z) \right] \quad \text{with} \quad G = G_P,$$

and proceed as above.

- With a finite *collection* of models $\mathcal{S}^* \subset \mathcal{S}$ we get a different target parameter Ψ_S for each $S \in \mathcal{S}^*$.
- With an infinite collection of models $\mathcal{S}^*$ (e.g., indexed by $\beta \in \mathbb{R}^p$) the previous approach is problematic.

$\Longrightarrow$ It is desirable to fit the weights *once* so that they are "universally" applicable for estimating the performance of all $S \in \mathcal{S}$.

One idea is to use undersmoothed HAL to do this.