**Main supplementary material of Phillips et al. 2023, Practical considerations for specifying a super learner**

**Contents**

Example 1: Super learner specification based on guidelines (pages 2–4)

Example 1a: Ensemble super learner specification (pages 2–3)

Example 1b: Cross-validated super learner as the discrete super learner (page 4)

Example 2: Poor super learner library specification (page 5)

Example 3: Poor cross-validation specification (page 6)

Example 4: Super learner specification for rare binary outcome based on guidelines (pages 7–8)

Example 5: Poor cross-validation specification with rare binary outcome (page 9)

Example 6: Super learner specification with complex real data, larger sample size, risk prediction application (pages 10–13)

Justification of the SL specification (page 11)

Limitations of the SL specification (page 11)

Results (pages 12–13)

Example 7: Super learner specification with complex real data, smaller sample size, causal inference application (pages 14–18)

Data structure (page 14)

Estimand, what are we aiming to learn from the data? (page 14)

Estimation, how will we approximate estimand from data? (pages 14–15)

Justification of the SL specification (page 16)

Limitations of the SL specification (page 16)

To estimate, or not to estimate, the propensity score in a randomized trial? (page 16)

Results (pages 16–18)

Computational considerations (page 19)

References (page 20)

**Example 1: Super learner specification based on guidelines**

We begin by illustrating the super learner (SL) specification with a simulated dataset. The simulation involves a set of four covariates, , each drawn from a uniform distribution . The outcome, , follows the function

,

where is the usual indicator function and is drawn from an independent standard normal distribution. A sample size of 50 was drawn to establish the analytic dataset. We will use the SL to learn a prediction function for the conditional mean of , given : . The R code to reproduce the analyses in examples 1–5 using the both the *SuperLearner* and *sl3* R packages is provided in supplemental files “Rcode\_SuperLearner.docx” and “Rcode\_sl3.docx”, respectively 1–3. Results from the *SuperLearner* implementation are reported here.

**Example 1a: Ensemble super learner specification**

The mean squared error (MSE) was chosen as the performance metric in the discrete SL (dSL) because the outcome is continuous. Observations are independent, so the effective sample size () equals the entire sample size, = 50. Thus, we specified V = 20 folds for the V-fold cross-validation (CV) scheme. The following learners were included in the library: mean; non-negative least squares (NNLS) regression; generalized linear models (GLM), including main terms GLM, GLM with two-way interactions among , and Bayesian GLM; elastic net regularized GLM regressions, including ridge and lasso regression; generalized additive model (GAM); and multivariate adaptive regression splines via enhanced adaptive regression through hinges (Earth). We constructed an ensemble SL (eSL) from this library using the default meta-learner, NNLS regression.

Supplemental Table S1. Results for the ensemble super learner (eSL) in example 1, part a, that used the default meta-learner, NNLS regression.

|  |  |  |
| --- | --- | --- |
| **Candidate learner in eSL** | **Cross-validated mean squared error** | **eSL coefficients** |
| Mean | 19.25 | 0 |
| Non-negative least squares regression | 25.80 | 0.07 |
| Generalized linear model (GLM) with main terms | 19.90 | 0 |
| GLM with two-way interactions | 25.36 | 0.01 |
| Bayesian GLM | 19.90 | 0 |
| Lasso regression | 17.97 | 0 |
| Ridge regression | 18.78 | 0 |
| Generalized additive model | 8.83 | 0.01 |
| Enhanced adaptive regression through hinges (Earth) | 4.77 | 0.91 |

The SL is defined by a function of the candidate learners included in the library, and the meta-learner’s job is to learn that function (Figure 2). Here, since NNLS was used as the meta-learner for the eSL the function of the candidates is a weighted combination of them, where the weights are constrained to be non-negative, sum to one, and estimated to minimize the MSE. As shown in Supplementary Table S1 above, the eSL is defined by a weighted combination of Earth, GAM, NNLS, and GLM with two-way interactions (denoted “GLM2” below). A prediction for this eSL (), given observed covariates (), is defined by

In what follows, similar representations of SL predictions omit the learners assigned zero coefficient, since they do not contribute to the SL.

From Supplementary Table S1, we can see that the NNLS candidate learner had the worst CV predictive performance, and it was assigned it a small weight of 0.07 in the eSL. Even though this learner performed poorly on its own, it produced a superior prediction function estimate (according to the NNLS meta-learner) when it was used it in combination with predictions from Earth, GAM, and GLM with two-way interactions. Earth had the best CV predictive performance, as its CV MSE was the lowest. If the dSL was given the library from part a, it would select learner Earth; specifically, a weighted combination, in which Earth is assigned a weight of one and all other candidates are assigned a weight of zero, would define the dSL. In part b of example 1, we show how we can augment the library for the dSL and consider the eSL as an additional candidate for the dSL, using the cross-validated SL (CV SL).

**Example 1b: Cross-validated super learner as the discrete super learner**

The CV SL provides the V-fold CV risk estimate for an eSL. It can be used as a dSL for evaluating an ensemble SL (eSL) and the candidates from which it was constructed. If the CV performance of the eSL is the best then the eSL fit from part a is the dSL, since the eSL has the best CV performance.

Supplemental Table S2. Example 1 results for the cross-validated super learner (CV SL) as the discrete super learner (dSL) for selecting the best-performing candidate learner from a library comprised of an ensemble SL (eSL) and the candidates from which the eSL was constructed.

|  |  |  |
| --- | --- | --- |
| **Candidate learner in dSL** | **Cross-validated mean squared error**a | **dSL coefficients**b |
| Mean | 17.21 | 0 |
| Non-negative least squares regression | 24.00 | 0 |
| Generalized linear model (GLM) with main terms | 18.97 | 0 |
| GLM with two-way interactions | 23.92 | 0 |
| Bayesian GLM | 18.97 | 0 |
| Lasso regression | 16.78 | 0 |
| Ridge regression | 17.50 | 0 |
| Generalized additive model | 8.34 | 0 |
| Enhanced adaptive regression through hinges (Earth) | 4.49 | 1 |
| eSL | 5.19 | 0 |

a The slight difference in CV risk results in parts a and b are due to variations in the cross-validation folds. b The dSL coefficients are provided in the standard output of the CV SL. They are provided here to show how the CV SL can be used as the dSL, and to distinguish them from the eSL coefficients.

Here, the CV MSE of the eSL was 5.19. Therefore, the eSL did not achieve lower CV risk than Earth. This highlights the importance of examining the CV risk of the eSL, or equivalently, considering the eSL as a candidate in a dSL. The eSL is not guaranteed to outperform its candidates. Based on Supplemental Table S2, we still should select Earth as the dSL. Thus, a prediction for the dSL (), given observed covariates (), is defined as .

**Example 2: Poor super learner library specification**

Example 2 mirrors example 1, aside from the library specification. The parametric learners from the example 1 library were considered in the SL library for example 2: mean; non-negative least squares (NNLS) regression; GLMs, including main terms GLM, GLM with two-way interactions among , and Bayesian GLM; and elastic net regressions, including ridge and lasso regression. Like example 1, we considered the eSL as an additional candidate in the dSL in example 2, by evaluating its CV MSE using CV SL.

Supplemental Table S3. Example 2 results for the cross-validated super learner (CV SL) as the discrete super learner (dSL) for selecting the best-performing candidate learner from a library comprised of an ensemble SL (eSL) and the parametric candidate learners from which the eSL was constructed.

|  |  |
| --- | --- |
| **Candidate learner in dSL** | **Cross-validated mean squared error** |
| Mean | 17.21 |
| Non-negative least squares regression | 24.00 |
| Generalized linear model (GLM) with main terms | 18.97 |
| GLM with two-way interactions | 23.92 |
| Bayesian GLM | 18.97 |
| Lasso regression | 16.78 |
| Ridge regression | 17.50 |
| eSL | 17.63 |

In example 2, the dSL is the mean learner. This example illustrates how the performance drops when the library lacks diversity in the learning strategies. Since the true functional form is unknown in real data examples, it is a good idea to consider a variety of learning strategies. Also, like example 1, the eSL in example 2 did not perform the best; the CV risk of some of the learners from which it was constructed (ridge and mean) was better.

**Example 3: Poor cross-validation specification**

Example 3 mirrors example 1, aside from the V-fold CV scheme. Here, we specified V-fold CV with V=2 folds.

Supplemental Table S4. Example 3 results for the cross-validated super learner (CV SL) as the discrete super learner (dSL) for selecting the best-performing candidate learner from a library comprised of an ensemble SL (eSL) and the candidates from which the eSL was constructed.

|  |  |
| --- | --- |
| **Candidate learner in dSL** | **CV mean squared error** |
| Mean | 18.44 |
| Non-negative least squares regression | 30.09 |
| Generalized linear model (GLM) with main terms | 33.14 |
| GLM with two-way interactions | 29.39 |
| Bayesian GLM | 33.12 |
| Lasso regression | 29.41 |
| Ridge regression | 25.73 |
| Generalized additive model | 17.44 |
| Enhanced adaptive regression through hinges (Earth) | 9.09 |
| eSL | 16.37 |

Examples 1 and 3 selected the same learner as their dSL, Earth. Example 3 CV risks are higher because V was specified too low with respect to the sample size. The CV MSE of Earth was doubled when V = 2 was considered vs. V = 20. The dSL selection of the same learner under different V should not be expected in real data; that is, this simulation study does not warrant consideration of lower V in the dSL! This illustrative example, and the previous one, are intended to emphasize the importance of the choices made by the analyst when they fit an algorithm to their data.

**Example 4: Super learner specification for a rare binary outcome based on guidelines**

Examples 4 and 5 consider a risk prediction task with almost the same data generating procedure (DGP) as examples 1–3, but with sample size of 5,000 a rare binary outcome. The rare binary outcome was derived from the continuous outcome in the previous DGP; we used an arbitrary cutoff of -8.75, which is close to the minimum value of the outcome. Specifically, outcomes less than -8.75 were set to one, and those greater than or equal to -8.75 were set to zero. This produced a rare binary outcome with only 41 events among the 5,000 samples.

The negative log-likelihood (NLL) was chosen as the performance metric in the discrete SL (dSL) because the outcome is binary, and this is a prediction task. The effective sample size () equals the minimum betweenand 5\*, where is the number of samples in the minority class, . Thus, we specified V = 20 folds for the V-fold CV scheme. Also, because the outcome is binary, we specified a stratified cross-validation scheme, so the outcome prevalence in training and validation sets could be like the entire analytic dataset’s outcome prevalence. The following learners were included in the library: mean; GLMs, including main terms GLM, GLM with two-way interactions among , and Bayesian GLM; lasso regression; GAM; and Earth. We constructed an eSL from this library using a meta-learner that minimizes the NLL. Like previous examples, we considered the eSL as an additional candidate in the dSL, by evaluating its CV NLL using CV SL. The results are presented in the table below.

Supplemental Table S5. Example 4 results for the cross-validated super learner (CV SL) as the discrete super learner (dSL) for selecting the best-performing candidate learner from a library comprised of an ensemble SL (eSL) and the candidates from which the eSL was constructed.

|  |  |
| --- | --- |
| **Candidate learner in dSL** | **CV negative log likelihood** |
| Mean | 0.048 |
| Generalized linear model (GLM) with main terms | 0.024 |
| GLM with two-way interactions | 0.024 |
| Bayesian GLM | 0.024 |
| Lasso regression | 0.024 |
| Generalized additive model | 0.013 |
| Enhanced adaptive regression through hinges (Earth) | 0.012 |
| eSL | 0.011 |

The eSL has the lowest CV NLL; it is therefore the dSL and we can use it to obtain predictions for the rare binary outcome in this hypothetical example. The eSL coefficients are as follows: 0.49 to GAM, 0.51 to Earth, and zero to all other candidates. Thus, a prediction for the dSL in example 4 (), given observed covariates (), is defined as

.

For binary outcomes the predictions are predicted probabilities. This example illustrates how to calculate the effective sample size with a binary outcome, and in the R code we show how one can choose a different performance metric and specify stratified V-fold CV.

**Example 5: Poor cross-validation specification with rare binary outcome**

Example 5 mirrors example 4, aside from the V-fold CV scheme. Here, we specified V-fold CV without consideration of the effective sample size. Since = 5000, we specified V = 5 folds for the V-fold CV scheme, and we did not specify a stratified CV procedure.

Supplemental Table S6. Example 5 results for the cross-validated super learner (CV SL) as the discrete super learner (dSL) for selecting the best-performing candidate learner from a library comprised of an ensemble SL (eSL) and the candidates from which the eSL was constructed.

|  |  |
| --- | --- |
| **Candidate learner in dSL** | **CV negative log likelihood** |
| Mean | 0.048 |
| Generalized linear model (GLM) with main terms | 0.024 |
| GLM with two-way interactions | 0.024 |
| Bayesian GLM | 0.024 |
| Lasso regression | 0.024 |
| Generalized additive model | 0.013 |
| Enhanced adaptive regression through hinges (Earth) | 5.495 |
| eSL | 0.014 |

Like example 4, the eSL has the lowest CV NLL in example 5; it is therefore the dSL. The eSL coefficients are as follows: 0.88 to GAM, 0.12 to Earth, and zero to all other candidates. (The cross-validated eSL was assigning similar weights to GAM and Earth.) Thus, a prediction for the dSL in example 5 (), given observed covariates (), is defined as .

Things can start breaking when the CV scheme does not consider because some training sets can have no outcomes. The huge increase in Earth’s CV risk in example 5 compared to example 4 is indicative that something has gone wrong. The NLL loss is supposed to be bounded between zero and one.

**Example 6: SL specification for larger sample size, risk prediction application**

In this example we will use the flowchart to specify a SL in a risk prediction application based on the International Stroke Trial (IST), a multinational randomized control trial (RCT) conducted from 1991 to 1996 that aimed to test the safety of aspirin and heparin in adults hospitalized for stroke. Patients suspected of suffering an ischemic stroke were randomized to aspirin or no aspirin, with and without heparin, to assess 14-day mortality (i.e., whether a patient died within 14 days of treatment randomization) and other outcomes of interest 4. We will consider a pre-processed version of the publicly available individual patient data from this trial 5. Our analytic dataset consists of 19,410 patients. The analytic dataset is available in file “data\_example6.csv” and the R code for data pre-processing is available in file “Rcode\_preprocess.docx”. The code to run the dSL in example 6 is provided in “Rcode\_sl3.docx”.

We are interested in predicting risk of 14-day mortality, given information measured at baseline, and we define baseline as the time that treatment randomization occurred. That is, we will use the SL to learn a prediction function for the conditional probability of 14-day mortality (), given covariates measured at baseline (): . There are a total of 24 covariates in the analytic dataset, including 3 continuous covariates, 6 binary covariates, and 15 categorical covariates. Of the 19,410 total patients, 1,686 died within 14 days of randomization to treatment () and 17,724 patients did not ().

Supplemental Table S7. Super learner (SL) specification for Example 6 based on the flowchart.

|  |  |
| --- | --- |
| Prediction function |  |
| Sample size, | = 19,410 |
| Dependent variable | Mortality |
| Predictors | 24 baseline covariables, including 3 continuous, 6 binary, and 15 categorical variables. |
| Performance metric for discrete SL | Negative log-likelihood |
| Effective sample size, |  |
| V-fold cross-validation (CV) scheme | Stratified V-fold CV (stratified by ) with 5 folds |
| Learners considered in the discrete SL library | GLM, Bayesian GLM, GAM, regularized elastic net GLMs, BART, RF, Earth, polynomial MARS, and HAL.  Additionally, an eSL that used the above learners in its library and NNLS meta-learner was considered as another candidate. |

**Justification of the SL specification**

This library is well-rounded in its learning strategies and can adapt to a diversity of true functional forms in a robust way. The is quite large, and the number of covariates is significantly smaller than so we were less concerned with screeners in this example but note that some of the algorithms reduce the covariate space as part of their procedure.We included complex machine learning algorithms that are capable of learning complicated, but potentially relevant, interactions. We also included a range of more simplistic parametric algorithms. We consider an eSL as a candidate, which offers the ability to further diversify the library for the dSL.

All aspects of training the learners that are based on the outcome-covariate relationships in the data must take place within the CV procedure, i.e., only on the data not left out during each CV loop. We confirm that this is true in our example. The following aspects of this setup were not based on the outcome-covariate relationships in the data: analytic dataset pre-processing; the selection of learners included in the library; and the selected learner tuning parameters.

**Limitations of the SL specification**

The library is missing is learners developed from subject-matter insight, and learners that have shown promise in related research. For instance, if it’s known that there are interactions among specific covariates, then those can be used to define learners which will incorporate them explicitly, e.g., via a formula. This library could also use more carefully tuned non-parametric machine learning algorithms, like gradient boosted trees.

This SL specification is limited in that it is very computationally expensive. We chose five folds for the V-fold CV scheme, the lowest number of recommended folds in Step 2 of the flowchart, to improve computational feasibility. We felt this was safe because is quite large. We ran the SL on a server overnight, and to examine how long this might take to run on an analyst’s laptop, we did not consider parallel processing to speed up the runtime. About 10 hours elapsed to run the SL. [Computational considerations for the SL are discussed in the next section, on page 13. The R session information for running the SL is available with the R code (see file “Rcode\_sl3.docx”).]

**Results**

Supplemental Table S8. Cross-validated (CV) negative log-likelihood (NLL) estimated for all candidates considered in the discrete super learner’s (dSL) estimation of the prediction function considered in example 6: the conditional probability of 14-day mortality (), given covariates measured at baseline ().

|  |  |  |
| --- | --- | --- |
| **Candidates** | **dSL coefficients** | **CV NLL** |
| Generalized linear model (GLM) | 0 | 0.242 |
| Bayesian GLM | 0 | 0.242 |
| Generalized additive model (GAM) | 0 | 0.242 |
| Lasso regression | 0 | 0.242 |
| Elastic net regression | 0 | 0.242 |
| Ridge regression | 0 | 0.242 |
| Enhanced adaptive regression through hinges (Earth) | 0 | 0.244 |
| Polynomial multivariate adaptive regression splines | 0 | 0.243 |
| Random forest | 0 | 0.247 |
| Bayesian additive regression trees | 0 | 0.739 |
| Highly adaptive lasso | 0 | 0.243 |
| Ensemble SL | 1 | 0.241 |

We can see that the dSL ended up selecting the eSL, as it has the lowest CV risk and therefore was assigned a coefficient of one, with all other learners assigned coefficients of zero. The dSL is defined as a weighted sum of the candidate, and therefore the dSL is identical to this learner, i.e., its predictions will be equal to the eSL’s predictions. The eSL predictions are a weighted combination of the candidates’ predictions. The eSL is defined by the following weighted combination of the candidates: weight of 0.28 to GAM, weight of 0.18 to PolyMARS, weight of 0.04 to Earth, weight of 0.25 to RF, weight of 0.25 to HAL, and weight of 0 to all the other candidates. Thus, a prediction for this dSL (), given observed covariates (), is defined as

.

Occasionally the candidate with the best performance (and therefore the learner selected as the dSL) is a complex algorithm (e.g., non-parametric machine learning algorithm) and its performance is only slightly better than a parametric candidate learner. For instance, as shown in the table above, the CV risk of the eSL was only slightly better than some of the candidates, like GAM and GLM. Even though the eSL has slightly better performance than the more simplistic algorithms, one still might consider proceeding with the GLM or GAM algorithm. Perhaps it offers a level of interpretability, explainability or transparency that is not provided by the eSL, and this is believed to outweigh the small decrease in performance. The evaluation of the CV risk that’s provided by the SL justifies this change to use the more simplistic algorithm, as this evidence that it performs similarly to the top-performing candidate.

In summary, the decision to work with any specific algorithm should be based on examination of its CV risk, along with the CV risk of a rich diversity of other algorithms. As considered in the hypothetical scenario above, the decision based on the CV risk estimates might be to proceed with a simpler algorithm instead of a complex algorithm. This is completely justifiable when the CV risk of the simpler algorithm is good. This notion also extends to complex algorithms, like the eSL. One should not proceed with the eSL until its CV risk, along with many other candidates’ CV risk, has been examined. Only at that point will the decision to work with the eSL be verified. This is exactly why we always recommend the dSL: it requires the estimation of candidate’s CV risk, generating the evidence that’s needed to decide which algorithm to use.

**Example 7: SL specification for smaller sample size, causal inference application**

In this example, we will use the guidelines to specify a SL in a causal inference application based on the Acupuncture for Chronic Headache in Primary Care (ACHPC) trial, a pragmatic randomized control trial (RCT) with non-blinded randomized treatment that was carried out in England and Wales from 1999 to 2001 to assess the effect of acupuncture in practice () on headache () 6. De-identified data were made publicly available by the study team 7. We will consider a pre-processed version of the publicly available individual patient data from this trial (see file “data\_example7.csv”). The R code for pre-processing is available in the file “Rcode\_preprocess.docx”. The code to run the SL in example 7 is provided in “Rcode\_sl3.docx”.

**Data structure**

A total of 401 adult general care patients having chronic headache were randomized to either to receive up to 12 treatments of acupuncture over three months (), or to receive usual care (). The outcome of interest () was headache score at 12 months. Only 301 of the 401 subjects completed the 12-month follow-up; The lost to follow-up variable will be denoted , with meaning was observed and meaning was not observed. We will also use notation to denote the covariates measured at baseline.

**Estimand (*what are we aiming to learn from our data?*)**

The target estimand in the ACHPC trial was the intention-to-treat (ITT) effect of acupuncture versus no acupuncture on 12-month headache score, i.e., the marginal additive treatment effect (ATE) among the population of interest. The statistical estimand is

,

where the outer expectation is taken over . The statistical interpretation of is the estimated average difference means across strata of. To interpret the estimate causally, as an effect of acupuncture versus no acupuncture on 12-month headache score, causal identifiability assumptions must be verified. Such an elaboration is beyond the scope of this example.

**Estimation (*how will we use the data to approximate the estimand?)***

We will estimate with targeted minimum loss-based estimation / targeted maximum likelihood estimation (TMLE) 8, accounting for LTFU. We will use SL for estimation of the missingness mechanism, which is the conditional probability that the outcome is observed given treatment and covariates, . We will also use SL for initial estimation of the outcome regression, which is the conditional mean outcome given treatment and covariates, , and this will be estimated using the data on 301 patients with no LTFU. The R code to reproduce the analysis uses the *sl3* R package and *tmle* R package 9 and is available in the file “Rcode\_sl3.docx”.

Supplemental Table S9. Super learner (SL) specifications based on the guidelines for SL estimation of the missingness mechanism and the outcome regression in example 7.

|  |  |  |
| --- | --- | --- |
| Prediction function |  |  |
| Sample size, | = 401 | = 301 |
| Dependent variable | Missingness indicator ()  denotes was observed and denotes was not observed | Headache score at 12 months () |
| Predictors | *The SL specifications for the missingness mechanism and the outcome regression considered the same predictor variables.*  , which is comprised of 26 variables, including 17 continuous , 7 binary , and binary . | |
| Performance metric for dSL | Negative log-likelihood | Mean squared error |
| Effective sample size, |  |  |
| V-fold cross-validation (CV) scheme | Stratified V-fold CV (stratified by ) with 20 folds | Stratified V-fold CV (stratified by ) with 20 folds |
| Learners considered in the discrete SL library | *The SL specifications for the missingness mechanism and the outcome regression considered the same library.*  The following learners were coupled with and without screeners and included as candidates: GLM, Bayesian GLM, GAM, regularized elastic net GLMs, BART, random forest (RF), Earth, polynomial MARS, and HAL.  Screeners included a lasso screener, to select the predictors with non-zero lasso coefficients; random forest screener, to select the top 15 most important predictors according to its variable importance metric; and a correlation screener, to select the predictors with a correlation test p-value less than 0.1. | |

**Justification of the SL specification**

This library is well-rounded in its learning strategies and can adapt to a diversity of true functional forms in a robust way. We coupled the learners with and without screeners to establish candidates less prone to overfitting. Also, some of the algorithms like BART reduce the covariate space as part of their procedure. We included complex machine learning algorithms that are capable of learning complicated, but potentially relevant, interactions. We also included a range of more simplistic algorithms.

All aspects of training the learners that are based on the outcome-covariate relationships in the data must take place within the CV procedure, i.e., only on the data not left out during each CV loop. We confirm that this is true in our example. The following aspects of this setup were not based on the outcome-covariate relationships in the data: analytic dataset pre-processing; the selection of learners included in the library; and the selected learner tuning parameters.

**Limitations of the SL specification**

The library is missing is learners developed from subject-matter insight, and learners that have shown promise in related research. For instance, if it’s known that there are interactions among specific covariates, then those can be used to define learners which will incorporate them explicitly, e.g., with a formula.

This SL specification is slightly computationally expensive due to the high number of folds. SL estimation of the missingness mechanism and the outcome regression took about 75 minutes each and no parallelization was considered. [Computational considerations for the SL are discussed towards the end of the supplement, on page 13. The R session information for running the SL is available with the R code (see file “Rcode\_sl3.docx”).]

**To estimate, or not to estimate, the propensity score in a randomized trial**

The propensity score (PS) is the conditional probability of receiving the treatment given covariates, . It has been proven that estimating the PS improves efficiency, i.e., decreases variance, compared to not estimating the PS. Even though the PS is known in an RCT, it would be nice to draw on that result and therefore we do recommend estimation of the PS. Since estimator consistency (i.e., unbiased estimation in large samples) is guaranteed in an RCT, it would be good to not stray too far from the known PS in this estimation. To preserve consistency while gaining some efficiency, we recommend a simple main terms logistic regression for PS estimation in an RCT. For this reason, we did not use SL for PS estimation in this example. Instead, we estimated the PS with a GLM and included all as regressors. Our recommendation should be handled with the analyst’s discretion.

**Results**

The SL results for example 7 are provided on the following pages, in Supplemental Table S10 and Supplemental Table S11. The learner selected by the dSL for estimation of the missingness mechanism was the correlation screener coupled with HAL, and the learner selected by the dSL for the outcome regression estimation was the RF screener coupled with lasso regression. The TMLE estimate of was -5.162 (95% confidence interval: -7.534, -2.799).

Supplemental Table S10. Cross-validated negative log-likelihood estimates for all candidates considered in the discrete super learner’s (dSL) estimation of the missingness mechanism in example 7. For candidates defined by screener-learner couplings, the screener is listed before plus symbol. The learner selected by the discrete SL is bolded.

|  |  |
| --- | --- |
| Lasso + GLMa | 0.587 |
| Lasso + Bayesian GLM | 0.575 |
| Lasso + GAMa | 0.588 |
| Lasso + Lasso | 0.582 |
| Lasso + Elasticnet.5 | 0.581 |
| Lasso + Ridge | 0.575 |
| Lasso + Polynomial MARSa | 0.589 |
| Lasso + Eartha | 0.595 |
| Lasso + RFa | 0.604 |
| Lasso + BARTa | 0.559 |
| Lasso + HALa 2-way interactions | 0.588 |
| Lasso + HAL 3-way interactions | 0.586 |
| RF + GLM | 0.591 |
| RF + Bayesian GLM | 0.564 |
| RF + GAM | 0.636 |
| RF + Lasso | 0.576 |
| RF + Elasticnet.5 | 0.574 |
| RF + Ridge | 0.559 |
| RF + Polynomial MARS | 0.582 |
| RF + Earth | 0.662 |
| RF + RF | 0.586 |
| RF + BART | 0.556 |
| RF + HAL 2-way interactions | 0.544 |
| RF + HAL 3-way interactions | 0.554 |
| Correlation + GLM | 0.578 |
| Correlation + Bayesian GLM | 0.568 |
| Correlation + GAM | 0.653 |
| Correlation + Lasso | 0.570 |
| Correlation + Elasticnet.5 | 0.570 |
| Correlation + Ridge | 0.558 |
| Correlation + Polynomial MARS | 0.584 |
| Correlation + Earth | 0.643 |
| Correlation + RF | 0.578 |
| Correlation + BART | 0.554 |
| **Correlation + HAL 2-way interactions** | **0.543** |
| Correlation + HAL 3-way interactions | 0.545 |
| GLM | 0.614 |
| Bayesian GLM | 0.573 |
| GAM | 0.672 |
| Lasso | 0.565 |
| Elasticnet.5 | 0.564 |
| Ridge | 0.560 |
| Polynomial MARS | 0.576 |
| Earth | 0.624 |
| RF | 0.582 |
| BART | 0.556 |
| HAL 2-way interactions | 0.580 |
| HAL 3-way interactions | 0.578 |

a generalized linear model, GLM; generalized additive model, GAM; multivariate adaptive regression splines, MARS; MARS via enhanced adaptive regression through hinges, Earth; random forest, RF; highly adaptive lasso, HAL; Bayesian additive regression trees, BART.

Supplemental Table S11. Cross-validated mean squared error estimates for all candidates considered in the discrete super learner’s (dSL) estimation of the outcome regression in example 7. For candidates defined by screener-learner couplings, the screener is listed before plus symbol. The learner selected by the discrete SL is bolded.

|  |  |
| --- | --- |
| Lasso + GLMa | 125.304 |
| Lasso + Bayesian GLM | 125.302 |
| Lasso + GAMa | 123.902 |
| Lasso + Lasso | 125.138 |
| Lasso + Elasticnet.5 | 125.192 |
| Lasso + Ridge | 125.990 |
| Lasso + Polynomial MARS | 122.802 |
| Lasso + Eartha | 131.193 |
| Lasso + RFa | 147.969 |
| Lasso + BARTa | 127.680 |
| Lasso + HALa 2-way interactions | 122.900 |
| Lasso + HAL 3-way interactions | 124.186 |
| RF + GLM | 121.736 |
| RF + Bayesian GLM | 120.622 |
| RF + GAM | 122.686 |
| **RF + Lasso** | **119.846** |
| RF + Elasticnet.5 | 120.654 |
| RF + Ridge | 120.769 |
| RF + Polynomial MARS | 120.313 |
| RF + Earth | 164.558 |
| RF + RF | 123.561 |
| RF + BART | 122.594 |
| RF + HAL 2-way interactions | 124.610 |
| RF + HAL 3-way interactions | 157.139 |
| Correlation + GLM | 121.663 |
| Correlation + Bayesian GLM | 121.661 |
| Correlation + GAM | 121.370 |
| Correlation + Lasso | 120.559 |
| Correlation + Elasticnet.5 | 121.332 |
| Correlation + Ridge | 121.800 |
| Correlation + Polynomial MARS | 120.892 |
| Correlation + Earth | 139.447 |
| Correlation + RF | 123.747 |
| Correlation + BART | 122.778 |
| Correlation + HAL 2-way interactions | 123.822 |
| Correlation + HAL 3-way interactions | 141.544 |
| GLM | 122.423 |
| Bayesian GLM | 121.198 |
| GAM | 131.194 |
| Lasso | 121.416 |
| Elasticnet.5 | 121.327 |
| Ridge | 120.661 |
| Polynomial MARS | 120.892 |
| Earth | 187.680 |
| RF | 126.040 |
| BART | 125.843 |
| HAL 2-way interactions | 127.128 |
| HAL 3-way interactions | 166.269 |

a generalized linear model, GLM; generalized additive model, GAM; multivariate adaptive regression splines, MARS; MARS via enhanced adaptive regression through hinges, Earth; random forest, RF; highly adaptive lasso, HAL; Bayesian additive regression trees, BART.

**Computational considerations**

Tailoring the SL specification based on the flowchart will robustify the SL, and this may often be at the expense of computational feasibility. When the library is designed to be able to adapt to a diversity of true functional forms in a robust way, the computational cost for running the SL will likely increase. This includes longer runtimes and greater memory allocation. Also, computational cost increases with the number of V-fold CV folds, V, and with sample size, .

There are a few ways to improve the computational feasibility of the SL without sacrificing performance. First, parallel processing can be used to speed up the computational runtime and the standard SL software provides options for this. Second, as reviewed in the manuscript’s subsection on the V-fold CV specification, V-fold CV schemes with smaller V can be considered for larger to improve computational feasibility. This intuition is reflected in Step 2 of the flowchart, where it is recommended that V be set to at least some minimum value, and this minimum decreases as increases. By setting V to the minimum recommended value, computational feasibility will be better than if it was any higher. Third, some experienced analysts might know how to pare down the library without sacrificing quality (e.g., by reducing the number of variants of the same learner with different tuning parameter specifications to a smaller combination of tuning parameters with reasonable ranges). Last, SLs that combine subset-specific candidate fits have been proposed and offer the potential to improve SL scalability for very large 10. This functionality is available in the *subsemble* R package.

In some cases, the analyst will not be able to run the SL on their machine, and server space will become a necessity. Additionally, the runtime for the SL might take a day or longer, so the analyst should be prepared for this. A rule of thumb with large datasets is to trial run the SL on a subset of the analytic dataset, without examining any results. This provides the opportunity to catch and fix issues that might arise when running the SL in a timely fashion.

The computational cost of the SL is high, but the time and funding toward study design and data curation is likely much greater. The study design, data curation, analysis and interpretation should be considered as a concerted effort; the results from these studies hinge on the rigor of the analysis and the interpretation.

**References**

1. R Core Team. R: A language and environment for statistical computing. 2022. https://R-project.org/. 9 November 2022.
2. Polley E, LeDell E, van der Laan M. SuperLearner: Super Learner Prediction. 2021. https://CRAN.R-project.org/package=SuperLearner. 9 November 2022.
3. Coyle J, Hejazi N, Malenica I, Phillips RV, Sofrygin O. sl3: Pipelines for Machine Learning and Super Learning. 2022. https://github.com/tlverse/sl3. 9 November 2022.
4. Group IS. The International Stroke Trial (IST): a randomised trial of aspirin, subcutaneous heparin, both, or neither among 19 435 patients with acute ischaemic stroke. *Lancet*. 1997 May 31;349(**9065**):1569-81.
5. Sandercock PA, Niewada M, Członkowska A. The international stroke trial database. *Trials*. 2011 Dec;12(**1**):1-7.
6. Vickers AJ, Rees RW, Zollman CE, McCarney R, Smith CM, Ellis N, Fisher P, Van Haselen R. Acupuncture for chronic headache in primary care: large, pragmatic, randomised trial. *BMJ*. 2004 Mar 25;328(**7442**):744.
7. Vickers AJ. Whose data set is it anyway? Sharing raw data from randomized trials. *Trials*. 2006 Dec;7(**1**):1-6.
8. Van der Laan MJ, Rose S. Targeted learning: causal inference for observational and experimental data. New York: Springer; 2011 Jun 6.
9. Gruber S, van der Laan MJ. tmle: An R package for targeted maximum likelihood estimation. *J Stat Softw*. 2012 Nov 16;51:1-35.
10. Sapp S, van der Laan MJ, Canny J. Subsemble: an ensemble method for combining subset-specific algorithm fits. *J Appl Stat.* 2014 Jun 3;41(**6**):1247-59.