The New England Journal of Statistics in Data Science Volume 0, 1–9 (2025)
DOI: https://doi.org/10.51387/25-NEJSDS82

Practical Considerations for Variable Screening in the Super Learner

Brian D. WILLIAMSON*, Drew KING, and Ying HUANG

Abstract

Estimating a prediction function is a fundamental component of many data analyses. The super learner ensemble, a particular implementation of stacking, has desirable theoretical properties and has been used successfully in many applications. Dimension reduction can be accomplished by using variable screening algorithms (screeners), including the lasso, within the ensemble prior to fitting other prediction algorithms. However, the performance of a super learner using the lasso for dimension reduction has not been fully explored in cases where the lasso is known to perform poorly. We provide empirical results that suggest that a diverse set of candidate screeners should be used to protect against poor performance of any one screener, similar to the guidance for choosing a library of prediction algorithms for the super learner. These results are further illustrated through the analysis of HIV-1 antibody data.

keywords and phrases: Super learner, Ensemble machine learning, Variable screening, Prediction.

1. INTRODUCTION

Estimating a prediction function is a fundamental component of statistical data analysis. Based on measured outcome Y and covariates X, the goal is to estimate the conditional expectation $E(Y \mid X)$. There are many approaches to estimating this regression function, ranging from simple and fully parametric [e.g., generalized linear models; 20] to flexible machine learning approaches, including random forests [3], gradient boosted trees [13], the lasso [27], and neural networks [2]. While a single estimator (also referred to as a learner) may be chosen, it can be advantageous to instead consider an ensemble of multiple candidate learners; a large ensemble of flexible learners increases the chance that one learner can approximate the underlying conditional expectation well.

The super learner (SL) [29, 24] is one such ensemble, and is related to stacking [32]. The super learner has been shown to have the same expected loss for predicting the outcome as the oracle estimator, asymptotically [29]. If both simple and complex algorithms are included in the library of candidate learners, the cross-validation used within the super learner to select the optimal combination of candidate learners to minimize a cross-validated loss function can minimize the risk of overfitting [1]. The super learner has been used successfully in many applications [see, e.g., 28, 23, 21, 18, 6] and is implemented in several software packages for the R programming language [25, 10].

In some settings, it may be of interest to perform variable selection as part of certain candidate learners within the super learner. This includes high-dimensional settings where prediction performance may be improved by reducing the dimension prior to prediction and settings where having a parsimonious set of variables is a goal of the analysis. While recent work has developed general guidelines for specifying a super learner [22], the choice of *screening algorithms* (often referred to as *screeners*) has been relatively unexplored. In particular, there are cases where theory suggests that the lasso does not consistently select the most relevant variables [17]. In this article, we explore the use of the lasso as a screener within a super learner ensemble, with the goal of determining if there are cases where the performance of the ensemble is sensitive to possible poor performance of the lasso screener.

2. OVERVIEW OF VARIABLE SCREENING IN THE SUPER LEARNER

Phillips et al. [22] provide a thorough overview of the super learner algorithm, which we briefly summarize here. The super learner takes as input the following: the dataset $\{(X_i, Y_i)\}_{i=1}^n$; a *library* of candidate learners (e.g., random forests, the lasso, neural networks), possibly including combinations with variable screeners (e.g., the lasso) that reduce the dimension of the covariates prior to prediction; a fixed number of cross-validation folds; and a loss function to minimize using cross-validation. The *ensemble super learner* (hereafter eSL) uses a meta-learner to combine the predictions from the candidate learners [22]. Below, we will refer to a special case of the eSL, which we call the *cSL*: the convex combination of the candidate learners that minimizes the cross-validated loss. The combination weights are

arXiv: 2311.03313
*Corresponding author.

greater than or equal to zero by definition. The discrete super learner (dSL) selects the single candidate learner that minimizes the cross-validated loss.

Including variable screeners in the SL library is motivated by the fact that reducing the number of covariates can improve prediction performance in some cases [see, e.g., 27], for example, high-dimensional settings. Screeners can be broadly categorized as outcome-blind, such as removing one variable from a pair of highly correlated covariates; or based on the outcome-covariate relationship. Examples of this latter category include removing covariates with univariate outcome-correlation-test p-value larger than a threshold; removing covariates with random forest variable importance measure [3] rank larger than a threshold; or removing covariates with zero estimated lasso coefficient.

Strategies based on the outcome-covariate relationship, if pursued, should be combined with other algorithms in the SL library and should be evaluated using cross-validation [22]. In practice, specifying a screener-learner combination results in a new learner, where first the screener is applied and then the learner is applied on the reduced set of covariates. This becomes one of the learners in the SL library, and like any other learner, can either be chosen as part of the optimal combination or assigned zero weight. For example, suppose that q screeners and ℓ learners are considered. Then the candidate library could consist of all $q \times \ell$ screener-learner combinations, or a subset of these combinations chosen by the analyst. Below, we will consider all $q \times \ell$ screener-learner pairs. The ensembling step of the super learner assigns non-negative coefficients to each of the screener-learner combinations to create the ensemble learner.

3. NUMERICAL EXPERIMENTS

3.1 Data-Generating Mechanisms

To demonstrate the performance of the SL procedure using different screeners, we consider several data-generating scenarios. In each scenario, our simulated dataset consists of independent replicates of (X, Y), where $X = (X_1, \ldots, X_p)$ is a covariate vector and Y is the outcome of interest.

We consider a continuous outcome with $Y \mid (X = x) = f(x) + \epsilon$, where $\epsilon \sim N(0, 1)$ independent of X; and a binary outcome with $Pr(Y = 1 \mid X = x) = \Phi\{f(x)\}$, where Φ denotes the cumulative distribution function of the standard normal distribution (so Y follows a probit model). The outcome regression function f is either linear, with $f(x) = x\beta$, or nonlinear, with

$$\begin{aligned} f(x) &= \beta_1 f_1\{c_1(x_1)\} + \beta_2 f_2\{c_2(x_2), c_3(x_3)\} \\ &\quad + \beta_3 f_3\{c_3(x_3)\} + \beta_4 f_4\{c_4(x_4)\} \\ &\quad + \beta_5 f_2\{c_5(x_5), c_1(x_1)\} + \beta_6 f_3\{c_6(x_6)\}, \\ f_1(x) &= \sin\left(\frac{\pi}{4}x\right), \quad f_2(x, y) = xy, \\ f_3(x) &= x, \quad f_4(x) = \cos\left(\frac{\pi}{4}x\right). \end{aligned}$$

The functions $c_1, \ldots, c_6$ scale each variable to have mean zero and standard deviation one. The vector β determines the strength of the relationship between outcome and covariates. We define a weak relationship between the outcome and covariates by setting $\beta = (0, 1, 0, 0, 0, 1, \mathbf{0}_{p-6})$, where $p - 6$ variables do not affect the outcome, and a stronger relationship between the outcome and covariates by setting $\beta = (-3, -1, 1, -1.5, -0.5, 0.5, \mathbf{0}_{p-6})$. The covariates follow a multivariate normal distribution with mean zero and covariance matrix Σ. In the uncorrelated case, Σ is the identity matrix. In the correlated case, the variables in the active set (a subset of the first six variables) have correlation 0.9 (in the case of the strong outcome-covariate relationship) or 0.95 (in the case of the weak relationship) while the remaining variables have correlation 0.3. Based on the strength of relationship between outcome and features, whether it is linear or nonlinear, and whether the features are correlated, the outcome rate in the binary case ranges from approximately 13% to 80%.

3.2 Prediction Algorithms

We compared several main prediction algorithms: the lasso, the cSL without including the lasso in its library of candidate learners [referred to as cSL (-lasso)], the cSL including the lasso (referred to as cSL), and the dSL with and without the lasso in its library of candidate learners (referred to as dSL and dSL (-lasso), respectively). For the super learner approaches, we further considered four possible sets of screeners that were fit prior to any learners: no screeners; a lasso screener only; rank correlation, univariate correlation, random forest, and lasso screeners (referred to as "All" screeners); and all possible screeners except the lasso [referred to as "All (-lasso)"]. Tuning parameters for the screeners depended on the total number of features, except for the lasso screener, which always removed variables with zero regression coefficient based on a tuning parameter selected by 10-fold cross-validation. For $p = 10$, we considered a screener that selected all variables and a univariate correlation screener that removed variables with outcome-correlation-test p-value less than 0.2. For $p > 10$, the rank correlation screeners removed variables outside of the top 10, 25, or 50 ranked correlation-test p-values; the univariate correlation screener removed variables with p-value less than 0.2 or 0.4; and the random forest screener removed variables outside of the top 10 or 25 most-important variables, ranked by the random forest variable importance measure [3].

We finalized our cSL specification following the guidelines specified in Phillips et al. [22]. First, because we were interested in estimating the true continuous prediction function for both continuous and binary outcomes, we estimated the V-fold cross-validated least squares loss (for continuous outcomes) or log-likelihood loss (for binary outcomes); we then

Table 1. All possible candidate learners for super learners used in the simulations, along with their R implementation, tuning parameter values, and description of the tuning parameters. All tuning parameters besides those listed here are set to their default values. In particular, the random forests are grown with a minimum node size of 5 for continuous outcomes and 1 for binary outcomes and a subsampling fraction of 1; the boosted trees are grown with shrinkage rate of 0.1, and a minimum of 10 observations per node.

Candidate learner	R implementation	Tuning parameter and possible values	Tuning parameter description
Generalized linear models	`base`	–	–
Random forests	`ranger` [33]	`num.trees` $= 1000$	Number of trees
		`min.node.size` $\in \{5, 20, 50, 100, 250\}$	Minimum node size
Gradient boosted trees	`xgboost` [7]	`max.depth` $= 4$	Maximum tree depth
		`ntree` $\in \{100, 500, 1000\}$	Number of iterations
		`shrinkage` $\in \{0.01, 0.1\}$	Shrinkage
Multivariate adaptive regression splines	`earth` [19]	`nk` $= \min\{\max\{21, 2p+1\}, 1000\}^{\dagger}$	Maximum number of model terms before pruning
Lasso	`glmnet` [12]	λ, chosen via 10-fold cross-validation	ℓ_1 regularization parameter

†: p denotes the total number of predictors.

used the non-negative least squares (NNLS) or non-negative log-likelihood metalearner to obtain the optimal convex combination of these learners, respectively. We used stratified cross-validation [16] in the binary-outcome case. We used nested cross-validation in all cases to estimate the performance of the cSL and all individual screener-learner pairs. Second, we computed the effective sample size n_{eff}, and based our choice of V on the flowchart in Figure 1 of Phillips et al. [22]; the values of V are provided below. Finally, our library of screener-learner pairs specified above was designed to be computationally feasible and adapt to high dimensions and different underlying true regression functions.

3.3 Experimental Overview

For each $n \in \{200, 500, 1000, 2000, 3000\}$, $p \in \{10, 500\}$, and simulation scenario described above, we generated 1000 random datasets according to this data generating mechanism. For continuous outcomes, $n_{\text{eff}} = n$; thus, we set $V = 20$ for $n \leq 500$ and set $V = 10$ otherwise. For binary outcomes, n_{eff} ranged from 10 (the 5% incidence outcome at $n = 200$) to 1367 (a 54% incidence outcome at $n = 3000$). We set $V = n_{\text{eff}}$ in three cases, and $V = 20$ or $V = 10$ otherwise, depending on the value of n_{eff}. The exact values of n_{eff} and V used are provided in the Supporting Information. We additionally generated a test dataset with sample size 1 million in each replication to estimate the true prediction performance of each prediction function estimated using V-fold cross-validation. We measured prediction performance for each algorithm described above using R-squared for continuous outcomes and area under the receiver operating characteristic curve (AUC) and non-negative log likelihood for binary outcomes. For the continuous outcome, R-squared is equivalent to the cross-validated metric that is being optimized: the mean squared error, which is equal to R-squared up to a scaling factor, the outcome variance. For the binary outcome, AUC is often of interest when assessing prediction performance. AUC is not equivalent to non-negative log-likelihood; however, developing a super learner using AUC loss can be unstable in some settings.

3.4 Results

We display the results under a strong outcome-feature relationship in Figures 1 and 2. Focusing first on a continuous outcome, when the outcome-feature relationship is linear (Figure 1 left column), all estimators have prediction performance converging quickly to the best-possible prediction performance as the sample size increases. In small samples with a linear relationship, removing the lasso from the SL library results in decreased performance. When the outcome-feature relationship is nonlinear (Figure 1 right column), the results depend on the variable screeners and algorithm used. The lasso has poor performance regardless of sample size, particularly in the case with correlated features; this is consistent with theory [17]. Also, particularly for large numbers of features (e.g., when $p = 500$), using the lasso screener alone within a super learner degrades performance, while using a large library of candidate screeners can improve performance over a super learner with no screeners. Having a large library of candidate screeners can protect against poor lasso performance. Results are similar for the binary outcome.

The results under a weak outcome-feature relationship follow similar patterns (Figures 3 and 4). In this case, the best-possible prediction performance is lower than in the strong-relationship case, as expected; and a larger sample size is required to achieve prediction performance close to this optimal level.

In the Supporting Information, we provide additional results. Results for the binary outcome with respect to non-negative log-likelihood follow similar patterns to those ob-

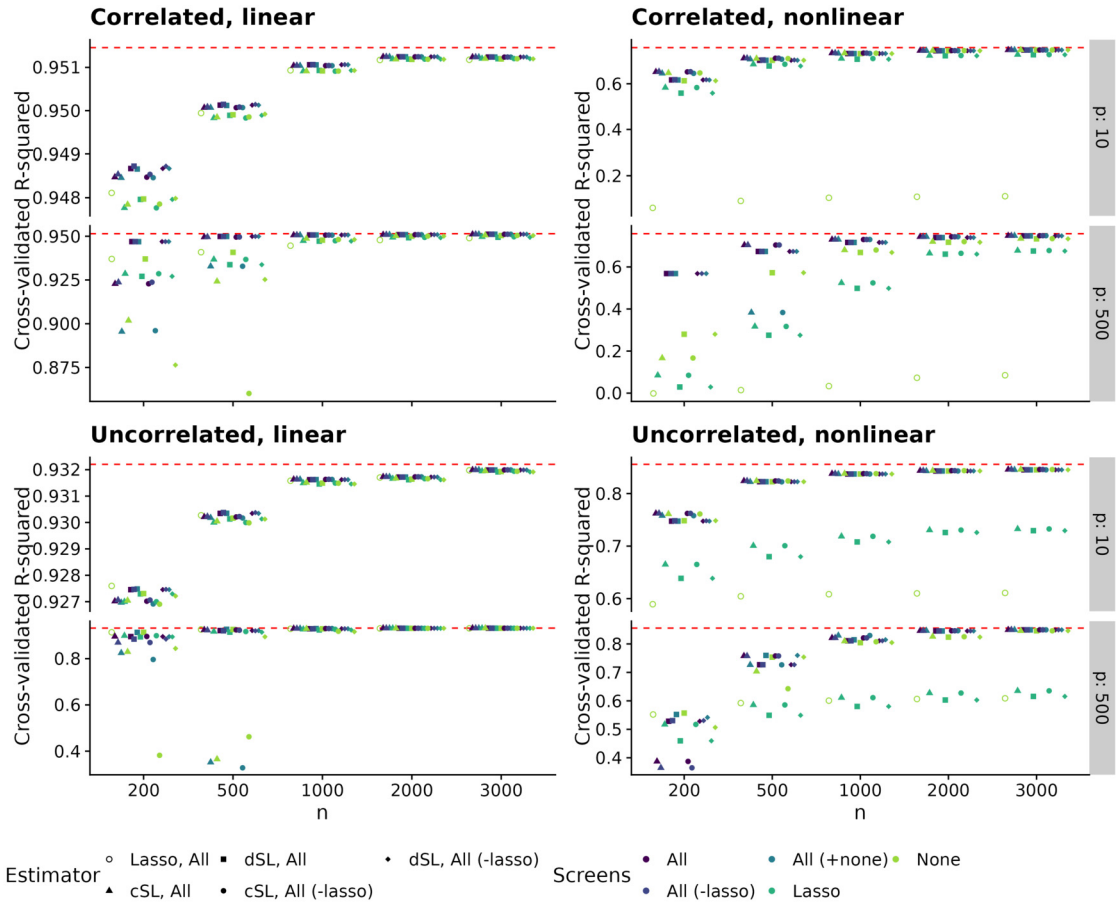

Figure 1: Prediction performance versus sample size n, measured using cross-validated R-squared, for predicting a continuous outcome. There is a strong relationship between outcome and features. The top row shows results for correlated features, while the bottom row shows results for uncorrelated features. The left-hand column shows results for a linear outcome-feature relationship, while the right-hand column shows results for a nonlinear outcome-feature relationship. The dashed line denotes the best-possible prediction performance in each setting. Color denotes the variable screeners, while shape denotes the estimator (lasso, convex ensemble super learner [cSL], and discrete super learner [dSL]). Note that the y-axis limits differ between panels.

served here using AUC. We considered further feature dimensions p with a fixed number of cross-validation folds V, and found similar results to the primary results presented above. Finally, we present results for $n = 500$ and $p = 2000$ and for candidate learners within the super learner. In the high-dimensional setting, performance follows the same trends across outcomes and estimators as the other (n, p) combinations.

4. PREDICTING HIV-1 NEUTRALIZATION SUSCEPTIBILITY

HIV-1 is a genetically diverse pathogen. Broadly neutralizing antibodies (bnAbs) against HIV-1 neutralize a wide array of HIV-1 genetic variants. One such bnAb, VRC01, was recently evaluated in two placebo-controlled randomized trials [9]. Predicting whether or not a given HIV-1 virus is susceptible to neutralization by a bnAb, including VRC01, is an important component of prevention research; several prediction models have been developed recently [15, 5, 14, 4, 26, 8, 34, 18, 30, 11, 31].

We analyze HIV-1 envelope (Env) amino acid (AA) sequence data from 611 publicly-available HIV-1 Env pseudoviruses made from blood samples of HIV-1 infected individuals [18]. In addition to binary indicators of specific AA residues at each position in the Env sequence, the data also include information on the geographic region of origin of the virus, the subtype of the virus, and viral geometry; there are over 800 features in total. We considered two outcomes of interest: the $\log_{10}$-transformed 50% inhibitory concentration, IC_{50}, defined as the concentration (µg/mL) of VRC01 necessary to neutralize 50% of viruses in vitro, with large values of IC_{50} indicating resistance to neutralization; and susceptibility to neutralization, defined as the binary indicator that

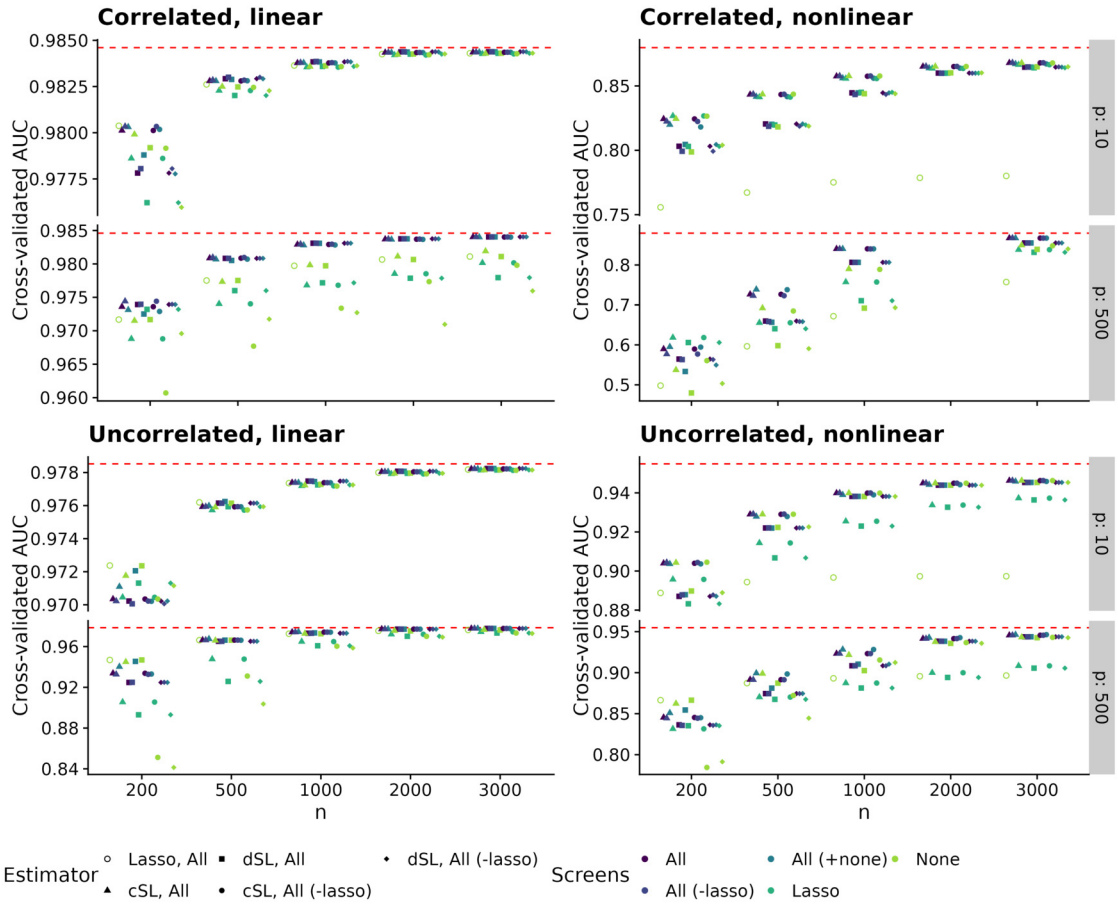

Figure 2: Prediction performance versus sample size n, measured using cross-validated AUC, for predicting a binary outcome. There is a strong relationship between outcome and features. The top row shows results for correlated features, while the bottom row shows results for uncorrelated features. The left-hand column shows results for a linear outcome-feature relationship, while the right-hand column shows results for a nonlinear outcome-feature relationship. The dashed line denotes the best-possible prediction performance in each setting. Color denotes the variable screeners, while shape denotes the estimator (lasso, convex ensemble super learner [cSL], and discrete super learner [dSL]). Note that the y-axis limits differ between panels.

$IC_{50} < 1$ µg/mL. For each outcome, we considered the same prediction algorithms and eSL specification as in Section 3. Following Phillips et al. [22], we set $V = 10$ for both the continuous and binary outcome.

The results are presented in Tables 2 and 3. For both outcomes, some screening tended to be beneficial. Among the analyses that used screeners, using the lasso screener alone resulted in the worst performance for the binary outcome and near the worst for the continuous outcome. Again, for both outcomes, having a large set of screeners protected against poor lasso performance; the lasso performed worse than the cSL or dSL for both outcomes. The lasso had a cross-validated (CV) R-squared for the continuous outcome of 0.331 with a 95% confidence interval (CI) of [0.305, 0.358], and a CV AUC for the binary outcome of 0.757 [0.633, 0.882]. For the continuous outcome, the largest point estimate of CV R-squared was achieved by the cSL with all screeners, including the lasso; the CV R-squared was 0.394 [0.371, 0.417]. The best-performing dSL was in the case with all screeners but the lasso, with CV R-squared 0.391 [0.372, 0.411]. For the binary outcome, the largest CV AUC for the cSL was 0.826 [0.723, 0.929] in the case with all screeners but the lasso; for the dSL, the largest CV AUC was 0.837 [0.737, 0.936] in the case with no screeners. In the Supporting Information, we present cross-validated performance for the candidate learners in each cSL; cross-validated negative log-likelihood loss for the binary susceptibility outcome; and the cSL coefficients and dSLs for each cross-validation fold.

5. DISCUSSION

In this manuscript, we explored the effect of using different combinations of variable screeners within the super learner. We found that both the lasso and the ensemble su-

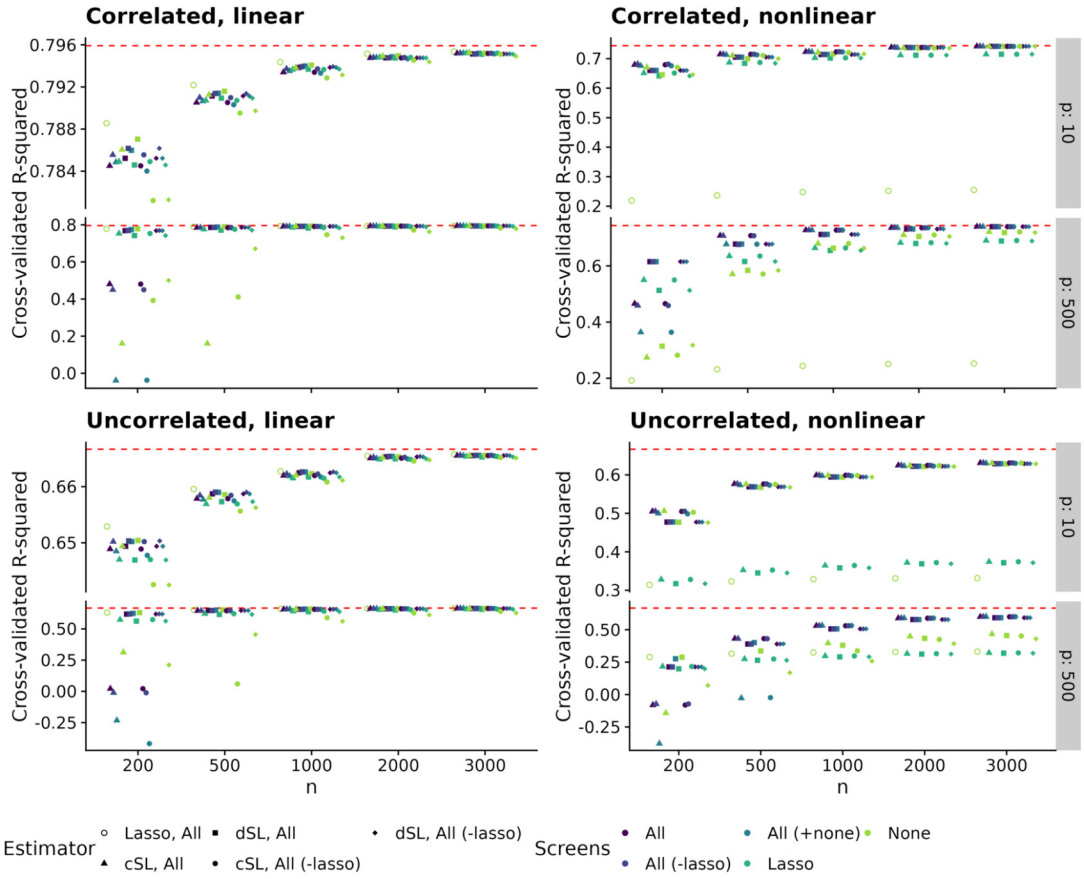

Figure 3: Prediction performance versus sample size n, measured using cross-validated R-squared, for predicting a continuous outcome. There is a weak relationship between outcome and features. The top row shows results for correlated features, while the bottom row shows results for uncorrelated features. The left-hand column shows results for a linear outcome-feature relationship, while the right-hand column shows results for a nonlinear outcome-feature relationship. The dashed line denotes the best-possible prediction performance in each setting. Color denotes the variable screeners, while shape denotes the estimator (lasso, convex ensemble super learner [cSL], and discrete super learner [dSL]). Note that the y-axis limits differ between panels.

per learner (cSL) using only a lasso screener had poor prediction performance when the outcome-feature relationship was nonlinear; in other words, in the case where the lasso is misspecified. However, if a sufficiently rich set of candidate screeners were included, then including the lasso as a candidate screener did not degrade performance. These results held for both continuous and binary outcomes, and for both strong and weak relationships between the outcome and features. The same patterns held for the discrete super learner (dSL). In an analysis of 611 HIV-1 envelope protein pseudoviruses with over 800 features, we found similar results to the simulations. There, the dSL tended to result in performance similar to the cSL.

Taken together, the results suggest that some caution must be used when specifying screeners within a super learner, but that a sufficiently large set of candidate screeners can protect against misspecification of a given screener. This guidance is similar to the guidance to specify a diverse set of learners in a super learner [22], and can be viewed as complementary, since an algorithm-screener pair defines a new candidate learner.

SUPPLEMENTARY MATERIAL

Additional numerical results are available in the Supporting Information. Code to reproduce all numerical experiments and the data analysis is available on GitHub at https://github.com/bdwilliamson/sl_screening_supplementary.

FUNDING

This work was supported by the National Institutes of Health (NIH) grants R01CA277133, R37AI054165, R01-GM106177, U24CA086368 and S10OD028685. The opinions expressed in this article are those of the authors and do not necessarily represent the official views of the NIH.

Accepted 20 March 2025

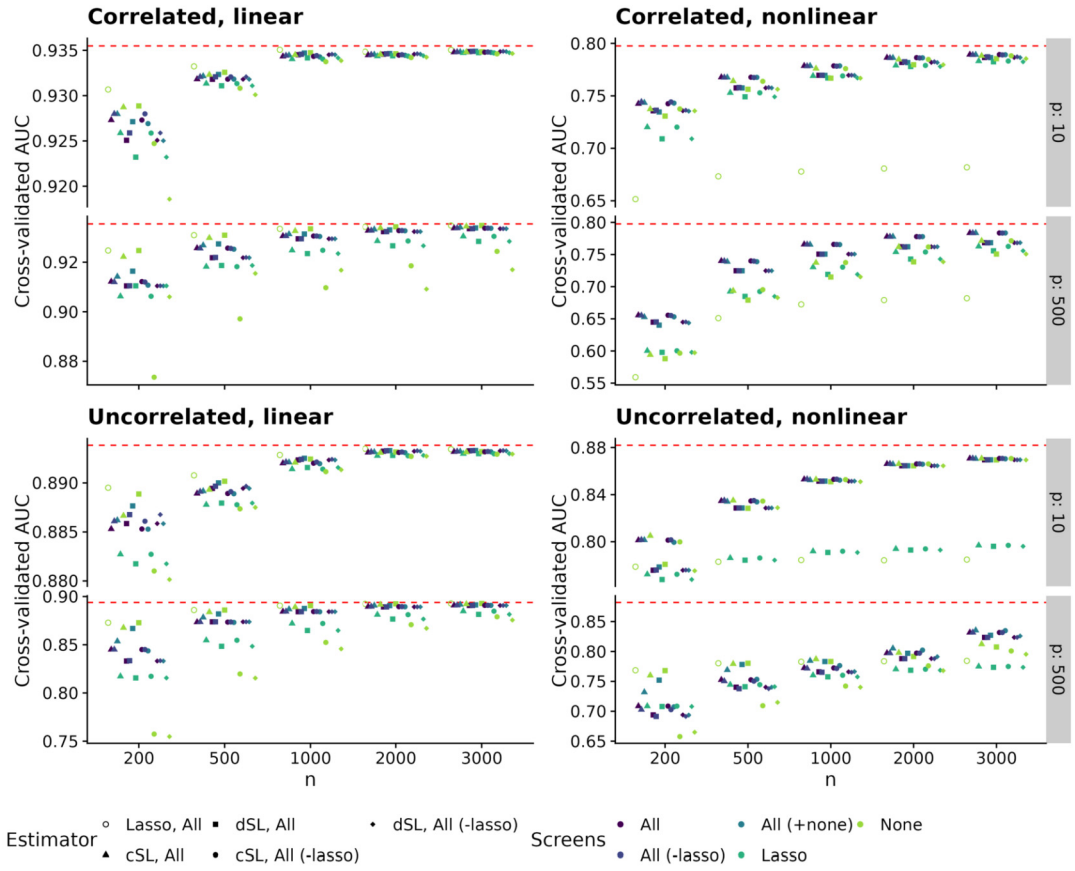

Figure 4: Prediction performance versus sample size n, measured using cross-validated AUC, for predicting a binary outcome. There is a weak relationship between outcome and features. The top row shows results for correlated features, while the bottom row shows results for uncorrelated features. The left-hand column shows results for a linear outcome-feature relationship, while the right-hand column shows results for a nonlinear outcome-feature relationship. The dashed line denotes the best-possible prediction performance in each setting. Color denotes the variable screeners, while shape denotes the estimator (lasso, convex ensemble super learner [cSL], and discrete super learner [dSL]). Note that the y-axis limits differ between panels.

Table 2. Estimates of cross-validated R-squared for the continuous IC_{50} outcome, for the convex ensemble super learner (cSL), the discrete super learner (dSL), and the lasso, under each combination of learners and screeners. For screeners, 'None' denotes no screeners; 'Lasso' denotes only a lasso screener; 'All (-lasso)' denotes random forest, rank-correlation, and correlation-test p-value screening; 'All' denotes these three screener types plus the lasso; and 'All (+none)' denotes all screeners plus the 'none' screener.

Learners	Screeners	Algorithm	Min	Max	Point estimate [95% CI]
All	None	cSL	0.208	0.501	0.373 [0.353, 0.393]
All	None	dSL	0.058	0.491	0.366 [0.347, 0.385]
All	None	lasso	0.331	0.331	0.331 [0.305, 0.358]
All	Lasso	cSL	0.175	0.527	0.388 [0.364, 0.414]
All	Lasso	dSL	0.173	0.516	0.387 [0.366, 0.409]
All	All (-lasso)	cSL	0.182	0.535	0.390 [0.370, 0.411]
All	All (-lasso)	dSL	0.192	0.519	0.391 [0.372, 0.411]
All	All	cSL	0.180	0.545	0.394 [0.371, 0.417]
All	All	dSL	0.173	0.516	0.387 [0.365, 0.409]
All	All (+none)	cSL	0.203	0.533	0.378 [0.354, 0.403]
All	All (+none)	dSL	0.173	0.516	0.387 [0.365, 0.409]

Table 3. Estimates of cross-validated AUC for the binary sensitivity outcome, for the convex ensemble super learner (cSL), the discrete super learner (dSL), and the lasso, under each combination of learners and screeners. For screeners, 'None' denotes no screeners; 'Lasso' denotes only a lasso screener; 'All (-lasso)' denotes random forest, rank-correlation, and correlation-test p-value screening; 'All' denotes these three screener types plus the lasso; and 'All (+none)' denotes all screeners plus the 'none' screener.

Learners	Screeners	Algorithm	Min	Max	Point estimate [95% CI]
All	None	cSL	0.755	0.874	0.823 [0.719, 0.928]
All	None	dSL	0.763	0.895	0.837 [0.737, 0.936]
All	None	lasso	0.647	0.813	0.757 [0.633, 0.882]
All	Lasso	cSL	0.727	0.865	0.806 [0.696, 0.915]
All	Lasso	dSL	0.730	0.897	0.811 [0.703, 0.919]
All	All (-lasso)	cSL	0.752	0.906	0.826 [0.723, 0.929]
All	All (-lasso)	dSL	0.772	0.907	0.827 [0.724, 0.929]
All	All	cSL	0.750	0.873	0.823 [0.719, 0.928]
All	All	dSL	0.772	0.897	0.826 [0.723, 0.929]
All	All (+none)	cSL	0.746	0.879	0.825 [0.720, 0.929]
All	All (+none)	dSL	0.772	0.897	0.829 [0.727, 0.931]

REFERENCES

[1] Balzer, L. B. and Westling, T. (2021). Demystifying statistical inference when using machine learning in causal research. *American Journal of Epidemiology* **200**.

[2] Barron, A. (1989). Statistical properties of artificial neural networks. In *Proceedings of the 28th IEEE Conference on Decision and Control* 280–285. IEEE.

[3] Breiman, L. (2001). Random forests. *Machine Learning* **45**(1) 5–32. MR3874153

[4] Bricault, C. A., Yusim, K., Seaman, M. S., Yoon, H., Theiler, J., Giorgi, E. E., Wagh, K., Theiler, M., Hraber, P., Macke, J. P., Kreider, E., Learn, G., Hahn, B., Scheid, J., Kovacs, J., Shields, J., Lavine, C., Ghantous, F., Rist, M., Bayne, M., Neubauer, G., McMahan, K., Peng, H., Cheneau, C., Jones, J., Zeng, J., Oschsenbauer, C., Nkolola, J., Stephenson, K., Chen, B., Gnanakaran, S., Bonsignori, M., Williams, L., Haynes, B., Doria-Rose, N., Mascola, J., Montefiori, D., Barouch, D. and Korber, B. (2019). HIV-1 neutralizing antibody signatures and application to epitope-targeted vaccine design. *Cell Host & Microbe* **25**(1) 59–72.

[5] Buiu, C., Putz, M. V. and Avram, S. (2016). Learning the relationship between the primary structure of HIV envelope glycoproteins and neutralization activity of particular antibodies by using artificial neural networks. *International Journal of Molecular Sciences* **17**(10) 1710.

[6] Carrell, D. S., Gruber, S., Floyd, J. S., Bann, M. A., Cushing-Haugen, K. L., Johnson, R. L., Graham, V., Cronkite, D. J., Hazlehurst, B. L., Felcher, A. H., Bejan, C. A., Kennedy, A., Shinde, M. U., Karami, S., Ma, Y., Stojanovic, D., Zhao, Y., Ball, R. and Nelson, J. C. (2023). Improving methods of identifying anaphylaxis for medical product safety surveillance using natural language processing and machine learning. *American Journal of Epidemiology* **192**(2) 283–295.

[7] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y. and Li, Y. (2019). xgboost: Extreme Gradient Boosting. R package version 0.82.1. https://CRAN.R-project.org/package=xgboost.

[8] Conti, S. and Karplus, M. (2019). Estimation of the breadth of CD4bs targeting HIV antibodies by molecular modeling and machine learning. *PLoS Computational Biology* **15**(4) 1006954.

[9] Corey, L., Gilbert, P. B., Juraska, M., Montefiori, D. C., Morris, L., Karuna, S. T., Edupuganti, S., Mgodi, N. M., DeCamp, A. C., Rudnicki, E. et al. (2021). Two randomized trials of neutralizing antibodies to prevent HIV-1 acquisition. *New England Journal of Medicine* **384**(11) 1003–1014. https://doi.org/10.1056/NEJMoa2031738.

[10] Coyle, J., Hejazi, N., Malencia, I., Phillips, R. and Sofrygin, O. (2023). sl3: Pipelines for machine learning and Super Learning. https://doi.org/10.5281/zenodo.1342293. https://github.com/tlverse/sl3.

[11] Dănăilă, V.-R. and Buiu, C. (2022). Prediction of HIV sensitivity to monoclonal antibodies using aminoacid sequences and deep learning. *Bioinformatics* **38**(18) 4278–4285.

[12] Friedman, J., Hastie, T. and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**(1) 1–22. https://doi.org/10.18637/jss.v033.i01.

[13] Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics* **29**(5) 1189–1232. https://doi.org/10.1214/aos/1013203451. MR1873328

[14] Hake, A. and Pfeifer, N. (2017). Prediction of HIV-1 sensitivity to broadly neutralizing antibodies shows a trend towards resistance over time. *PLoS Computational Biology* **13**(10) 1005789. https://doi.org/10.1371/journal.pcbi.1005789.

[15] Hepler, N. L., Scheffler, K., Weaver, S., Murrell, B., Richman, D. D., Burton, D. R., Poignard, P., Smith, D. M. and Kosakovsky Pond, S. L. (2014). IDEPI: rapid prediction of HIV-1 antibody epitopes and other phenotypic features from sequence data using a flexible machine learning platform. *PLoS Computational Biology* **10**(9) 1003842.

[16] Kohavi, R. (1996) *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. Stanford University ProQuest Dissertations Publishing.

[17] Leng, C., Lin, Y. and Wahba, G. (2006). A note on the lasso and related procedures in model selection. *Statistica Sinica* **16** 1273–1284. MR2327490

[18] Magaret, C., Benkeser, D., Williamson, B., Borate, B., Carpp, L. et al. (2019). Prediction of VRC01 neutralization sensitivity by HIV-1 gp160 sequence features. *PLoS Computational Biology* **15**(4) 1006952.

[19] Milborrow, S. (2021). earth: Multivariate Adaptive Regression Splines. R package version 5.3.1. https://CRAN.R-project.org/package=earth.

[20] Nelder, J. and Wedderburn, R. (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A* **135**(3) 370–384.

[21] Petersen, M. L., LeDell, E., Schwab, J., Sarovar, V., Gross, R., Reynolds, N., Haberer, J. E., Goggin, K., Golin, C., Arnsten, J., Rosen, M. I., Remien, R. H., Etoori, D., Wilson, I. B., Simoni, J. M., Erlen, J. A., van der Laan, M. J., Liu, H. and Bangsberg, D. R. (2015). Super learner analysis of electronic adherence data improves viral prediction and may provide strategies for selective HIV RNA monitoring. *JAIDS Journal of Acquired Immune Deficiency Syndromes* **69**(1) 109–118.

[22] Phillips, R. V., van der Laan, M. J., Lee, H. and Gruber, S. (2023). Practical considerations for specifying a super learner. *International Journal of Epidemiology* **52**(4) 1276–1285.

[23] Pirracchio, R., Petersen, M. L., Carone, M., Rigon, M. R., Chevret, S. and van der Laan, M. J. (2015). Mortality prediction in intensive care units with the Super ICU Learner Algorithm (SICULA): a population-based study. *The Lancet Respiratory Medicine* **3**(1) 42–52.

[24] Polley, E. C. and van der Laan, M. J. (2010). Super Learner in Prediction.

[25] Polley, E., LeDell, E., Kennedy, C. and van der Laan, M. (2021). SuperLearner: Super Learner Prediction. R package version 2.0-28. https://CRAN.R-project.org/package=SuperLearner.

[26] Rawi, R., Mall, R., Shen, C.-H., Farney, S. K., Shiakolas, A., Zhou, J., Bensmail, H., Chun, T.-W., Doria-Rose, N. A., Lynch, R. M., Mascola, J. R., Kwong, P. D. and Chuang, G.-Y. (2019). Accurate prediction for antibody resistance of clinical HIV-1 isolates. *Scientific Reports* **9**(1) 14696. https://doi.org/10.1038/s41598-019-50635-w.

[27] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **58**(1) 267–288. MR1379242

[28] van der Laan, M. and Rose, S. (2011) *Targeted Learning: Causal Inference for Observational and Experimental Data*. Springer Science & Business Media. https://doi.org/10.1007/978-1-4419-9782-1. MR2867111

[29] van der Laan, M., Polley, E. and Hubbard, A. (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology* **6**(1) 25. https://doi.org/10.2202/1544-6115.1309. MR2349918

[30] Williamson, B. D., Magaret, C. A., Gilbert, P. B., Nizam, S., Simmons, C. and Benkeser, D. (2021). Super LeArner Prediction of NAb Panels (SLAPNAP): a containerized tool for predicting combination monoclonal broadly neutralizing antibody sensitivity. *Bioinformatics* **37**(22) 4187–4192.

[31] Williamson, B. D., Magaret, C. A., Karuna, S., Carpp, L. N., Gelderblom, H. C., Huang, Y., Benkeser, D. and Gilbert, P. B. (2023). Application of the SLAPNAP statistical learning tool to broadly neutralizing antibody HIV prevention research. *iScience* **26**(9).

[32] Wolpert, D. (1992). Stacked generalization. *Neural Networks*

5(2) 241–259.

[33] Wright, M. N. and Ziegler, A. (2017). ranger: a fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software* **77**(1) 1–17. https://doi.org/10.18637/jss.v077.i01. MR4583337

[34] Yu, W.-H., Su, D., Torabi, J., Fennessey, C. M., Shiakolas, A., Lynch, R., Chun, T.-W., Doria-Rose, N., Alter, G., Seaman, M. S. et al. (2019). Predicting the broadly neutralizing antibody susceptibility of the HIV reservoir. *JCI Insight* **4**(17).

Brian D. Williamson. Kaiser Permanente Washington Health Research Institute, Fred Hutchinson Cancer Center, and University of Washington, USA.
E-mail address: brian.d.williamson@kp.org

Drew King. Seattle Central College, USA.

Ying Huang. Fred Hutchinson Cancer Center and University of Washington, USA.