Published in final edited form as:
Stat Appl Genet Mol Biol. 2007 ; 6: Article7.

Super Learning: An Application to the Prediction of HIV-1 Drug Resistance*

Sandra E. Sinisi[*], **Eric C. Polley**[†], **Maya L. Petersen**[‡], **Soo-Yon Rhee**[**], and **Mark J. van der Laan**[††]

[*] *University of California, Berkeley, sinisi54@alum.berkeley.edu*

[†] *University of California, Berkeley, ecpolley@berkeley.edu*

[‡] *University of California, Berkeley, mayaliv@gmail.com*

[**] *Stanford University, syrhee@stanford.edu*

[††] *Division of Biostatistics, School of Public Health, University of California, Berkeley, laan@stat.berkeley.edu*

Abstract

Many alternative data-adaptive algorithms can be used to learn a predictor based on observed data. Examples of such learners include decision trees, neural networks, support vector regression, least angle regression, logic regression, and the Deletion/Substitution/Addition algorithm. The optimal learner for prediction will vary depending on the underlying data-generating distribution. In this article we introduce the "super learner", a prediction algorithm that applies any set of candidate learners and uses cross-validation to select between them. Theory shows that asymptotically the super learner performs essentially as well as or better than any of the candidate learners. In this article we present the theory behind the super learner, and illustrate its performance using simulations. We further apply the super learner to a data example, in which we predict the phenotypic antiretroviral susceptibility of HIV based on viral genotype. Specifically, we apply the super learner to predict susceptibility to a specific protease inhibitor, nelfinavir, using a set of database-derived non-polymorphic treatment-selected mutations.

Keywords

cross-validation; loss-based estimation; machine learning; genomics; antiretroviral

1 Introduction

Numerous methods exist to learn from data the best predictor of a given outcome. A few examples include decision trees, neural networks, support vector regression, least angle regression, logic regression, and the Deletion/Substitution/Addition (D/S/A) algorithm. Such algorithms, or learners, can be characterized by the mechanism used to search the parameter space. For example, the D/S/A algorithm (Sinisi and van der Laan, 2004) uses polynomial basis functions, while logic regression (Ruczinski et al., 2003) constructs Boolean expressions of binary covariates. The relative performance of a given learner depends on how extensively

[*]Maya Petersen is supported by a Pre-Doctoral Fellowship from the Howard Hughes Medical Institute. Mark van der Laan is supported by NIH grant R01 GM071397. The authors wish to acknowledge Dr. Robert Shafer of Stanford University School of Medicine for making the example dataset available, and for his helpful comments.

each learner must search over subspaces of the parameter space (reflected in the variance) in order for the mechanism employed to achieve a comparable approximation of the truth (reflected in the bias). Thus, the relative performance of various learners will depend on the true data-generating distribution. In practice, it is generally impossible to know *a priori* which learner will perform best for a given prediction problem and data set.

The framework for unified loss-based estimation (van der Laan and Dudoit, 2003) suggests a solution to this problem in the form of a new estimator, which we call the "super learner." This estimator is itself a prediction algorithm, which applies a set of candidate learners to the observed data, and chooses the optimal learner for a given prediction problem based on cross-validated risk. Theoretical results show that such a super learner will perform asymptotically as well or better than any of the candidate learners (van der Laan and Dudoit, 2003; van der Laan et al., 2004). We present the super learner in the context of unified loss-based estimation in Section 2, and illustrate its performance in the context of a known data-generating distribution and varying sample sizes using a simulated example in Section 3.

In Section 4, we apply the super learner to research drawn from the treatment of Human Immunodeficiency Virus Type 1 (HIV-1). HIV frequently develops resistance to the antiretroviral drugs being used to treat it, resulting in loss of viral suppression and therapeutic failure. While over 15 licensed antiretroviral drugs exist, the majority fall into three classes: protease inhibitors (PIs), nucleoside reverse transcriptase inhibitors (NRTIs), and non-nucleoside reverse transcriptase inhibitors (NNRTIs). There is a high-level of cross-resistance within drug classes; a virus that has developed resistance to one drug in a class may also be resistant to other drugs in the same class. Thus, selecting a new "salvage" drug regimen for an individual who has developed resistance to his or her current regimen is not straightforward. Improved understanding of the genetic basis of resistance to specific antiretroviral drugs has the potential to guide selection of an effective salvage regimen.

In the data example presented in this paper, the goal is to relate mutations in the genes encoding the HIV-1 enzyme protease to changes in *in vitro* susceptibility to a specific antiretroviral drug of the protease inhibitor class, nelfinavir (NFV). The outcome of interest is phenotypic drug susceptibility, and the predictors consist of protease mutations. In previous work, Rhee et al. (2006) applied six different learning methods to predict phenotypic drug susceptibility based on viral genotype (the presence or absence of mutations): (1) decision trees, (2) neural networks, (3) support vector regression, (4) linear regression, and (5) least angle regression. Here, we apply the super learner to the dataset used by Rhee et al. (2006), using least angle regression, linear regression, the D/S/A algorithm, logic regression, ridge regression, and classification and regression trees as candidate learners. Some of these algorithms were chosen for inclusion due to their popularity for prediction applications (e.g. linear regression), while others were chosen based on their compatibility with the use of a large set of binary predictors (e.g. logic regression). We aimed to pick a set of learners ranging from the simple (e.g. main term linear regression) to learners which themselves are data-adaptive and can be fine-tuned using cross-validation (e.g. the D/S/A). We also propose convex combinations of the candidate learners. We note, however, that this is just a sample of the types of learning algorithms which could be applied.

2 Methods

2.1 Loss-Based Estimation

Super learning is based on unified loss-based estimation theory, as introduced in van der Laan and Dudoit (2003). We provide a brief description of this estimation road map before introducing the super learner.

Van der Laan and Dudoit (2003) provide a general framework for parameter estimation problems. The data consist of n i.i.d. realizations of random variables, $X_1, \ldots, X_n$, from an unknown data generating distribution, P_0. The goal is to use the data to estimate a parameter ψ_0 of the distribution P_0, where ψ_0 is defined as some function of P_0. That is, we wish to obtain an estimator, or function of the data, $\hat{\psi}$, that is close (in risk distance) to the parameter ψ_0. For example, in our HIV-1 data example, Y denotes a continuous measurement of viral drug susceptibility, and W is a d-dimensional vector of binary variables, each indicating the presence or absence of a mutation. X_i consists of the pair $X_i = (W_i, Y_i)$, measured for a viral sequence i. The parameter of interest ψ_0 corresponds to the conditional expected value of drug susceptibility Y given the mutation profile W.

The general strategy for loss-based estimation is driven by the choice of a loss function and relies on cross-validation for estimator selection and performance assessment. The proposed estimation road map can be stated in terms of the following three main steps (van der Laan and Dudoit, 2003).

1. *Definition of the parameter of interest in terms of a loss function.* For the full data structure, define the parameter of interest as the minimizer of the expected loss, or risk, for a loss function chosen to represent the desired measure of performance (e.g., mean squared error in regression).
2. *Construction of candidate estimators based on a loss function.* Define a finite collection of candidate estimators for the parameter of interest.
3. *Cross-validation for estimator selection and performance assessment*. Use cross-validation to estimate risk based on the observed data loss function and to select an optimal estimator among the candidates in Step 2.

In the regression setting, our parameter of interest is $E(Y|W)$, which we denote $\psi(W)$. The loss function for our parameter of interest is the squared error loss function, $(Y - \psi(W))^2$. More generally, one can define a loss function $L:(X,\psi) \rightarrow L(x,\psi) \in \Re$ as any function which maps a candidate parameter value ψ and observation X into a real number, and whose expected value (i.e., risk) is minimized at the parameter value $\psi_0 = \Psi(P_0)$ corresponding to the data generating distribution P_0.

We will use various learning methods to construct candidate estimators needed for Step 2, and then use cross-validation as described in Step 3 to choose the optimal estimator among the candidates. We propose a super learner to perform Steps 2 and 3.

2.2 Candidate Learning Algorithms

Least Angle Regression (**LARS**) (Efron et al., 2004) is a model selection algorithm available in the lars() package of R (http://www.r-project.org). *Logic Regression* (Ruczinski et al., 2003) is an adaptive regression methodology that attempts to construct predictors as Boolean combinations of binary covariates available in the LogicReg() package of R. The *Deletion/ Substitution/Addition (D/S/A) algorithm* (**D/S/A**) (Sinisi and van der Laan, 2004) for polynomial regression data-adaptively generates candidate predictors as polynomial combinations of continuous and/or binary covariates. It is available as an R package at http://www.stat.berkeley.edu/users/laan/Software/. *Classification and Regression Trees* (**CART**) (Breiman et al., 1984) is available in the rpart() package of R. *Ridge Regression* (Hoerl and Kennard, 1970) is available in the MASS package of R. All of these methods have the option to carry out selection using v-fold cross-validation. The selected fine-tuning parameter(s) can include the ratio of the $L1$ norm of the coefficient vector in LARS; the number of logic trees and leaves in Logic Regression; and the number of terms and a complexity measure on each of the terms in D/S/A.

2.3 The Cross-Validation Selector

Cross-validation divides the available *learning* set into a *training* set and a *validation* set. Observations in the training set are used to construct (or *train*) the estimators, and observations in the validation set are used to assess the performance of (or *validate*) these estimators. The cross-validation selector selects the learner with the best performance on the validation sets. In υ-fold cross-validation, the learning set is divided into υ mutually exclusive and exhaustive sets of as nearly equal size as possible. Each set and its complement play the role of the validation and training sample, respectively, giving υ splits of the learning sample into a training and corresponding validation sample. For each of the υ splits, the estimator is applied to the training set, and its risk is estimated with the corresponding validation set. For each estimator/learner the υ risks over the υ validation sets are averaged resulting in the so-called *cross-validated risk*. The estimator with the minimal cross-validated risk is selected.

2.4 Super Learner

It is helpful to consider each learner as an algorithm applied to empirical distributions. Thus, if we index a particular learner with an index k, then this learner can be represented as a function $P_n \rightarrow \hat{\Psi}_k(P_n)$ from empirical probability distributions P_n to functions of the covariates. Consider a collection of $K(n)$ learners $\hat{\Psi}_k$, $k = 1, \ldots, K(n)$., in parameter space $\boldsymbol{\Psi}$. The super learner is a new estimator defined as

$$\widehat{\Psi}(P_n) \equiv \widehat{\Psi}_{\widehat{K}(P_n)}(P_n),$$

where $\hat{K}(P_n)$ denotes the cross-validation selector described above, which simply selects the learner which performed best in terms of cross-validated risk. Specifically,

$$\widehat{K}(P_n) \equiv \arg\min_k E_{B_n} \sum_{i, B_n(i)=1} (Y_i - \widehat{\Psi}_k(P^0_{n,B_n}))^2,$$

where $B_n \in \{0, 1\}^n$ denotes a random binary vector whose realizations define a split of the learning sample into a training sample $\{i : B_n(i) = 0\}$ and validation sample $\{i : B_n(i) = 1\}$. Here P^1_{n,B_n} and P^0_{n,B_n} are the empirical probability distributions of the validation and training sample, respectively.

Under the Assumption A1 that the loss function is uniformly bounded, and the Assumption A2 that the variance of the ψ_0-centered loss function $L(X, \psi) - L(X, \psi_0)$ can be bounded by its expectation uniformly in ψ, van der Laan et al. (2004) (Theorem 2) establish the following finite sample inequality:

Theorem 1 (van der Laan et al. (2004))

Let $\{\hat{\psi_k} = \hat{\Psi}_k(P_n), k = 1, \ldots, K(n)\}$ *be a given set of* $K(n)$ *estimators of the parameter value* $\psi_0 = \arg\min_{\psi \in \Psi} \int L(x, \psi) dP_0(x)$. *Let* $d_0(\psi, \psi_0) \equiv E_{P_0}\{L(X, \psi) - L(X, \psi_0)\}$ *denote the risk difference between a candidate estimator* ψ *and the parameter* ψ_0. *Suppose that* $\hat{\Psi}_k(P_n) \in \boldsymbol{\Psi}$ *for all k, with probability 1. Let* $\widehat{K}(P_n) \equiv \arg\min_k E_{B_n} \int L(x, \widehat{\Psi}_k(P^0_{n,B_n})) dP^1_{n,B_n}(x)$ *be the* cross-validation selector, *and let* $\tilde{K}(P_n) \equiv \arg\min_k E_{B_n} \int L(x, \widehat{\Psi}_k(P^0_{n,B_n})) dP_0(x)$ *be the comparable* oracle selector. *Then, under assumptions A1 and A2, one has the following finite sample inequality for any* $\lambda > 0$ *(where* $C(\lambda)$ *is a constant, defined in van der Laan et al. (2004)):*

$$Ed_0(\widehat{\Psi}_{\widehat{K}(P_n)}(P^0_{n,B_n}), \psi_0) \leq$$

$$(1+2\lambda) Ed_0(\widehat{\Psi}_{\tilde{K}(P_n)}(P^0_{n,B_n}), \psi_0) + 2C(\lambda) \frac{1 + log(K(n))}{np}$$

Note that the oracle selector is defined in Theorem 1 as the estimator, among the $K(n)$ learners considered, which minimizes risk under the true data-generating distribution. In other words, the oracle selector is the best possible estimator given the set of candidates considered; however, it depends on both the observed data and P_0, and thus is unknown.

Applied to the super learner, Theorem 1 shows us that the the super learner performs as well (in terms of expected risk difference) as the oracle selector, up to a typically second order term. Thus, as long as the number of learners considered ($K(n)$) is polynomial in sample size, we conclude that the super learner is the optimal learner in the following sense:

- If, as is typical, none of the candidate learners (nor, as a result, the oracle selector) converge at a parametric rate, the super learner performs asymptotically as well (in the risk difference sense) as the oracle selector, which chooses the best of the candidate learners.
- If one of the candidate learners happens to search among a subspace that contains the truth, and thus achieves a parametric rate of convergence, then the super learner achieves the almost parametric rate of convergence log n/n.

We refer the interested reader to technical reports by van der Laan and Dudoit (2003) and van der Laan et al. (2004), as well as a more recent publication by van der Vaart et al. (2006), for an extended development of these theoretical results.

In our applications of the super learner, we use a collection of six candidate learners based on the following algorithms: LARS; D/S/A; Logic Regression; CART; linear regression; and Ridge Regression. The super learner is used to select the learner from among these candidates which performs best in terms of cross-validated risk. Figure 1 shows a depiction of the super learner estimation process.

3 Simulation

We applied the super learner to a simulated data set with a known data generating distribution. The following model was used to generate 500 observations ($i = 1, \dots, 500$):

$$y_i = 2w_1w_{10} + 4w_2w_7 + 3w_4w_5 - 5w_6w_{10} + 3w_8w_9 + N(0,1), \quad (1)$$

where $w_j \sim Bin(0.4)$, $j = 1, \dots, 10$. The data for a given observation thus consist of a 10-dimensional vector of covariates W, and an outcome Y. These 500 observations constitute the learning set.

We applied the super learner with 10-fold cross-validation to the learning set to estimate $E(Y|W)$. This involved partitioning the learning set into 10 parts. Each part in turn served as the validation set, while the other 9/10ths of the data served as the training set. The super learner applied the following set of candidate learners to each of the 10 training sets: LARS, linear regression, D/S/A, Logic Regression, CART, and Ridge Regression. For linear regression and LARS, two sets of input variables were used. One consisted of all main terms, and the other consisted of all main terms $w_1, \dots, w_{10}$ and all two-way interactions $w_1, \dots, w_{10}, w_1w_2, \dots, w_9w_{10}$. Internal (10-fold) cross-validation was used to select the optimal fraction in LARS. Internal 10-fold cross-validation was also used to select the fine-tuning parameters for each candidate Logic Regression and D/S/A learner:

- Logic Regression
- — trees $\in \{1, \dots, 5\}$
- — leaves $\in \{1, \dots, 20\}$
- D/S/A

- — terms ∈ {1, … , 10}
- — order-of-interactions ∈ {1, 2}

Application of each candidate learner to the 10 training sets yielded a set of 10 estimators for each candidate learner; in the case of Logic Regression, LARS, and D/S/A these optimal estimators were indexed by fine-tuning parameters selected using internal cross-validation. The cross-validated risk for each of these candidate estimators was estimated by evaluating each estimator applied to the corresponding validation set. The resulting cross-validated risks for each estimator averaged across validation sets are displayed in Table 1.

Based on the table of cross-validated risks, D/S/A and Logic Regression were identified as the top learners. Table 2 shows a more detailed comparison of these two learners, illustrating variation (in the case of Logic Regression) in the selection of fine-tuning parameters within distinct partitions of the learning set and the associated cross-validated risks. 10-fold cross-validation within each of the 10 training sets consistently selected 5 trees for Logic Regression. However, the number of leaves selected for Logic Regression and the number of terms selected by D/S/A varied across the 10 training sets. The winning learner between these two competitors also varied across partitions of the learning set, with the lowest-cross validated risk achieved sometimes by D/S/A and sometimes by Logic Regression. On average, however, Logic Regression outperformed D/S/A (average cross-validated risk of 1.043 versus 1.055, respectively). Thus, the super learner selected Logic Regression as the optimal learner.

As the winning learner, Logic Regression was then applied to the entire learning sample. The final logic tree is displayed in Figure 2 and can be written as:

```
−3.09 * ((not w9) or (not w8)) +4.58 * ((not w10) or (not w6)) +4.17*
(((not w6) and w6) or (w7 and w2)) −3.09 * ((not w5) or (not w4))
+0.839 * w1
```

This fit has an R^2 of 0.874.

Even though the super learner did not select D/S/A as the optimal learner, given the close competition between Logic Regression and D/S/A, we also applied D/S/A to the learning sample. The final D/S/A fit had nine terms with eight two-way interactions, and an R^2 of 0.913:

$$\widehat{y}=0.087 - 4.906w_6w_{10}+4.211w_2w_7+3.205w_8w_9+3.107w_4w_5 +1.984w_1w_{10} - 0.406w_7w_8 - 0.359w_6+0.406w_3w_6 - 0.325w_9w_{10} \quad (2)$$

This can be compared to the true model which had 5 two-way interaction terms. All 5 of these interaction terms were included in the final 9 term D/S/A fit, with coefficients extremely comparable to those of the true model.

To assess the performance of the two estimators, we simulated 5000 observations from the true model to generate an independent test set. We evaluated the performance of the candidate estimators (Figure 2 for the final Logic Regression fit and Equation 2 for the final D/S/A fit) on this set of 5000 observations. The Logic Regression fit yielded a mean squared prediction error (MSPE) of 1.37 with an R2 of 0.84 while the D/S/A fit yielded a MSPE of 1.05 with an R2 of 0.88.

Next, we applied the super learner to a dataset of increasing sample size. Specifically, we used the same data-generating experiment to generate 3 additional samples, of sizes n=100, n=1000, and n=10,000. The super learner, based on the same candidate learners, was applied to each of these datasets. The resulting estimated cross-validated risks are presented in Tables (3)-(5). As anticipated, the estimated cross-validated risks of the candidate learners vary less as sample

size increases. Clearly, variability in estimated cross-validated risk can affect the candidate learner selected when the super learner is applied to finite samples. Also, both D/S/A and Logic regression are converging at a parametric rate to the true model.

Finally, we applied the super leaner to a different data generating distribution where each candidate learner would not be converging at a parametric rate to the true model. The following model was used to generate 500 observations ($i = 1, \dots, 500$):

$$y_i = 2w_1w_{10} + 4w_2w_7 + 3w_4w_5 - 5w_6w_{10} + 3w_8w_9 + w_1w_2w_4 - 2w_7(1 - w_6)w_9 - 4(1 - w_{10})w_1(1 - w_4) + N(0,1), \quad (3)$$

where $w_j \sim Bin(0.4)$, $j = 1, \dots, 10$. The superleaner was applied using the same candidates and parameters as the first simulation. We focus our attention on the logic regression and D/S/A candidate learners.

We also propose a new candidate learner built as a convex combination of the other candidate learners. For example, let $\hat{y}_i$ be the predicted value for the i^{th} candidate learner. Then the convex learner between logic regression and D/S/A is $\hat{y}_{convex,\,\alpha} = \alpha\,\hat{y}_{DSA} + (1 - \alpha)\,\hat{y}_{Logic}$

We applied this convex learner using a large set of α values, each representing a new candidate learner. The final D/S/A model had 13 terms and the final Logic Regression model had 5 trees and 19 leaves. As we expected, none of the candidate learners found the true model. Table 6 contains the cross-validated risks for the candidate learners. Here we see D/S/A outperform Logic Regression in terms of lowest cross-validated risk, but that the convex combination of the two models outperforms both. The super learner selected the convex combination of Logic Regression and D/S/A with $\alpha = 0.8316$.

We also simulated a new data set with 1 million observations to calculate the true risk on the 3 candidate learners (Logic Regression, D/S/A, and the convex combination of the two). Table 7 contains the true risk for the models selected based on cross-validated risk above. We see that the convex model has the smallest true risk.

4 Data Analysis

A description of the data used in our analysis is available in Rhee et al. (2006). The HIV-1 sequences were obtained from publicly available isolates in the Stanford HIV Reverse Transcriptase and Protease Sequence Database. We focus on predicting viral susceptibility to protease inhibitors (PIs) based on mutations in the protease region of the viral strand.

Mutations were defined as amino acid differences from the subtype B consensus wild type sequence at positions 1–99 in protease. We used a subset of these mutations, the non-polymorphic treatment-selected mutations (TSMs), as predictors. The TSMs were previously identified as those significantly associated with antiretroviral therapy in persons infected with subtype B viruses (Rhee et al., 2005, 2006). The association of these mutations with previous treatment is thought to result from selection due to their contributions to resistance, suggesting that this is a promising set of candidate predictor variables. The 58 TSMs used, occurring at 34 positions in protease, are listed in Table 8. Mutations are referred to by position followed by amino acid substitution; for example, 90M refers to the occurrence of methionine at position 90.

The outcome of interest was standardized log fold change in drug susceptibility, where fold change was defined as the ratio of IC_{50} of an isolate to IC_{50} of a standard wildtype control isolate. IC_{50} is the concentration of a drug needed to inhibit viral replication by 50% where IC stands for *inhibitory concentration*. We applied our super learner to predicting susceptibility

to a single PI, nelfinavir (NFV). A mutation profile and corresponding NFV susceptibility was available for 740 viral isolates; this constituted the learning sample.

4.1 Super Learner Results

We applied the super learner with 10-fold cross-validation to select the optimal learner given the following set of candidates: LARS, Logic Regression, D/S/A, CART, Ridge Regression, and a linear regression fit including all 58 mutations as main terms. We found no difference in risk when using D/S/A to search over 1-way or 2-way interactions. Similarly, Rhee et al. (2006) found that including all two-way interactions among the mutations as input variables did not improve the prediction accuracy. Therefore, D/S/A did not consider interactions and used 10-fold cross-validation to select between 1 and 50 main terms.

Table 9 shows the estimated cross-validated risks of the candidate learners. The main term linear regression fit yielded estimators with the lowest average risk, 0.187, although the average cross-validated risk for D/S/A was comparable at 0.188. Based on these risk estimates, main term linear regression was selected as the optimal learner, and a main term linear regression model was fit using all 740 observations. Tables 10 and 11 display the super learner estimator, in this case the linear regression model of all main terms, fit on the entire learning sample.

Due to the similarity in cross-validated risk of linear regression and D/S/A, we also applied D/S/A to the learning sample. Cross-validation selected a final D/S/A estimator with 40 main terms. This can be contrasted to the super learner estimator, corresponding to the linear regression estimator with 58 main terms. The similarity in cross-validated risk between these two estimators suggests that prediction is only marginally improved by including the other 18 mutations in the prediction model.

We investigated the size of the mutation set needed to achieve a predictor with comparable cross-validated risk by examining the cross-validated risks for the best main term model of each size (as fit by D/S/A). The resulting plot, shown in Figure 3, illustrates that the decline in cross-validated risk flattens sharply as the regression models reach approximately 20 main terms. This suggests that while the truly optimal predictor may use all 58 treatment-selected mutations as main terms, the majority of the predictive information can be captured by much smaller models of around 20 main terms.

We examined the best model of each size selected by D/S/A in order to investigate which mutations provide the most predictive information. The best models of each size selected by D/S/A happened to be nested. For example, the best model of size 1 contained the mutation 90M, the best model of size 2 contained the mutations 90M and 30N, the best model of size 3 contained the mutations 90M, 30N, and 54V, etc. The best models of size 1–20 are summarized in Table 12.

The *p*-values for the coefficients from the linear regression fit (Tables 10,11) and the list of the best D/S/A models of each size (Table 12) provide alternative rankings of the importance of each candidate mutation for resistance to NFV. The two approaches produce quite comparable insight into the set of mutations key to predicting susceptibility to NFV; 18 of the 20 mutations selected in the D/S/A models of size 1–20 were among the top 20 mutations as ranked by *p*-value in the linear regression model. Both rankings agreed relatively well with existing understanding of the effect of protease mutations on phenotypic susceptibility to NFV; the website of the Stanford HIV Drug Resistance Database (hivdb.stanford.efu) provides a review of the relevant literature, as well as drug-specific resistance scores for each mutation intended to summarize this literature. Our analysis found that the top 20 predictors in both the D/S/A and linear regression models included the majority of mutations known to contribute significantly to NFV resistance, including the mutations 90M, 30N, 88S, and 84A. Only 2 of

the mutations which were our top predictors are not currently recognized as NFV resistance mutations (50L and 74S). In contrast, of the 38 mutations not included in our top 20, 17 are not considered to be associated with resistance to NFV, while the majority of the remainder are believed associated with only minor resistance. Examples of this group include mutations not currently known to be associated with resistance to any of the PI drugs (e.g. 79A and 58E) and mutations known to contribute to resistance to other PI drugs, but not thought to contribute to NFV resistance (e.g, 50V and 32I).

Examination of the mutations selected by the super learner estimator provides some insight into the role of individual mutations in affecting phenotypic drug resistance. However, we wish to emphasize that the goal of the current data analysis is to learn the optimal *predictor* of phenotypic resistance based on genotype, rather than to estimate the effect or to rank the contribution of each of a set of candidate mutations on phenotypic resistance. If our primary goal was to rank the importance of each of the candidate mutations, a better approach would be to apply, e.g., variable importance measures, as presented in (van der Laan (2006)) and applied in Birkner and van der Laan (2005). Such variable importance analysis would require examination of the pairwise correlations between mutations. However, while perhaps interesting in their own right, such correlations are not directly relevant to the goal of learning an optimal predictor, and thus are not examined here. Specifically, the inclusion of two highly correlated mutations in, for example, the linear regression model, can result in instability in and difficulty interpreting the corresponding coefficients, but will not affect the overall performance of the predictor.

5 Discussion

We have presented a super learner that uses cross-validation to select an optimal learner from among a set of candidate learners. Theoretical results show that the super learner will asymptotically outperform any of the candidate estimators it employs as long as the number of candidate learners is polynomial in sample size (or, if one of the candidate estimators it employs achieves a parametric rate of convergence, the super learner will converge at an almost parametric rate). These results suggest that the investigator pays a very small price for considering multiple alternative learners. Currently, most researchers employ one, or at most a handful, of learning algorithms to answer prediction questions. A better approach would be to apply as many candidate learners as are feasible given time and computing limitations, and choose among them using the super learner. While recognizing that computing time does pose a real constraint on the learners considered, as the simulation and data analysis results presented suggest, we have found it feasible to apply several learners, some of which, such as the D/S/A and Logic Regression, made use of an additional layer of cross-validation when selecting fine tuning parameters. The use of convex combinations of candidate learners did not add to the computational burden, since the results from each candidate learner were already available. We demonstrated in the second simulation the benefit of using convex combinations as additional candidate learners.

Of course, in practical applications using finite samples, there is no guarantee that the super learner will always select the optimal learner for a given data application. Our first simulation results illustrate this point well. Logic Regression was selected by the super learner as optimal (based on sample size n=500), with a slightly lower average cross-validated risk than the D/S/A algorithm. However, when the performance of the two estimators was evaluated on an independent test set, D/S/A slightly outperformed the Logic Regression estimator. Further, variability in the estimates of cross-validated risk for each of the candidate learners clearly depends on the size of the dataset. As a result, demonstrated in this article using simulations, the candidate learner selected can shift with increasing sample size. These results suggest that when employing the super learner, it is worthwhile to evaluate not only the final estimator

provided by the winning learner but also competitive estimators. The results of the data example further reinforce the utility of considering alternative learners with risks comparable to that of the optimal learner to provide additional insight into the data.

References

Birkner, MD.; van der Laan, MJ. Technical Report 196, Division of Biostatistics. University of California; Berkeley: Nov. 2005. Application of a Variable Importance Measure Method to HIV-1 Sequence Data. URL http://www.bepress.com/ucbbiostat/paper196/

Breiman, L.; Friedman, JH.; Olshen, RA.; Stone, CJ. The Wadsworth Statistics/Probability series. Wadsworth International Group; 1984. Classification and Regression Trees.

Efron B, Hastie T, Johnstone I, Tibshirani R. Least Angle Regression. The Annals of Statistics 2004;32 (2):407–499.

Hoerl AE, Kennard RW. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 1970;12(3):55–67.

Rhee S, Fessel WJ, Zolopa AR, Hurley L, Liu T, Taylor J, Nguyen DP, Slome S, Klein D, Horberg M, Flamm J, Follansbee S, Schapiro JM, Shafer RW. HIV-1 Protease and Reverse-Transcriptase Mutations: Correlations with Antiretroviral Therapy in Subtype B Isolates and Implications for Drug-Resistance Surveillance. The Journal of Infectious Disease 2005;192:456–465.

Rhee S, Taylor J, Wadhera G, Ravela J, Ben-Hur A, Brutlag D, Shafer RW. Genotypic Predictors of Human Immunodeficiency Virus Type 1 Drug Resistance. Proceedings of the National Academy of Sciences USA. 2006In Press

Ruczinski I, Kooperberg C, LeBlanc M. Logic Regression. Journal of Computational and Graphical Statistics 2003;12(3):475–511.

Sinisi, SE.; van der Laan, MJ. Deletion/Substitution/Addition Algorithm in Learning with Applications in Genomics. Statistical Applications in Genetics and Molecular Biology. 2004. URL http://www.bepress.com/sagmb/vol3/iss1/art18/. Article 18

van der Laan, MJ.; Dudoit, S. Technical Report 130, Division of Biostatistics. University of California; Berkeley: Nov. 2003 Unified Cross-Validation Methodology for Selection Among Estimators and a General Cross-Validated Adaptive Epsilon-Net Estimator: Finite Sample Oracle Inequalities and Examples. URL http://www.bepress.com/ucbbiostat/paper130/

van der Laan, MJ.; Dudoit, S.; van der Vaart, AW. Technical Report 142, Division of Biostatistics. University of California; Berkeley: February 2004. The Cross-Validated Adaptive Epsilon-Net Estimator. URL http://www.bepress.com/ucbbiostat/paper142/

van der Laan MJ. Statistical inference for variable importance. International Journal of Biostatistics 2006;2(1):2.

van der Vaart AW, Dudoit S, van der Laan MJ. Oracle inequalities for mulit-fold cross vaidation. Statistics and Decisions 2006;1

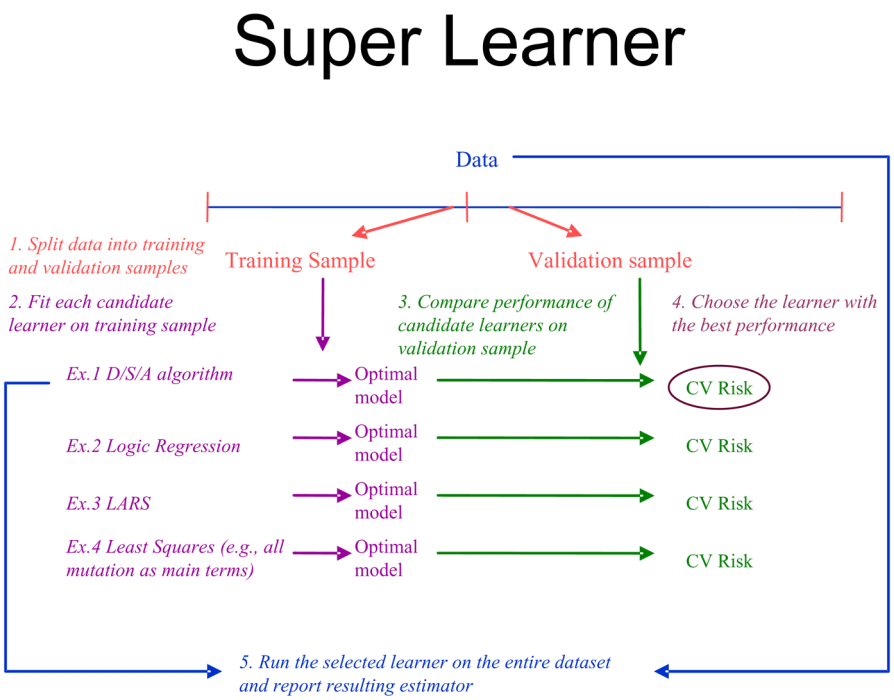

Figure 1.
Schematic Diagram of the Super Learner

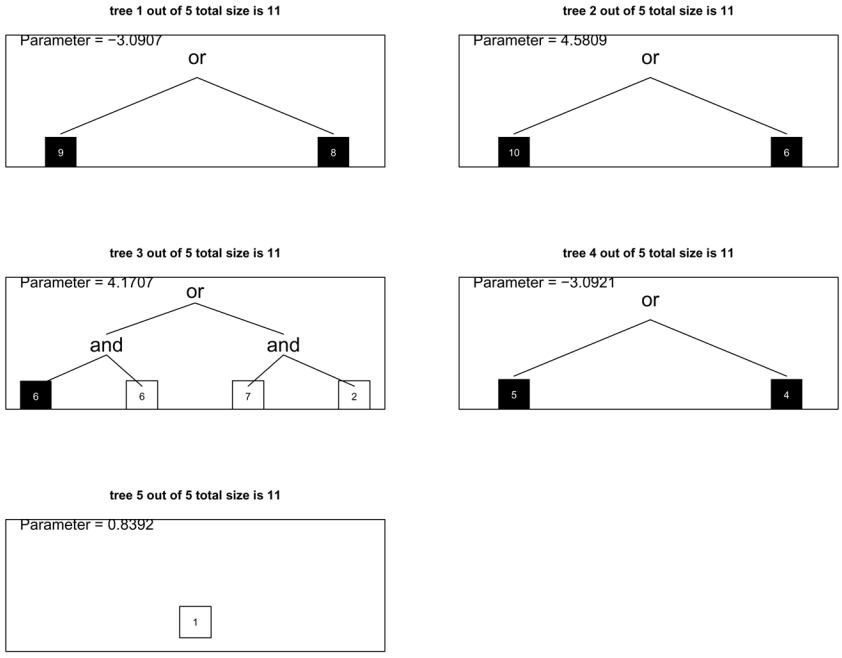

Figure 2. Simulated Example
Logic Regression Fit

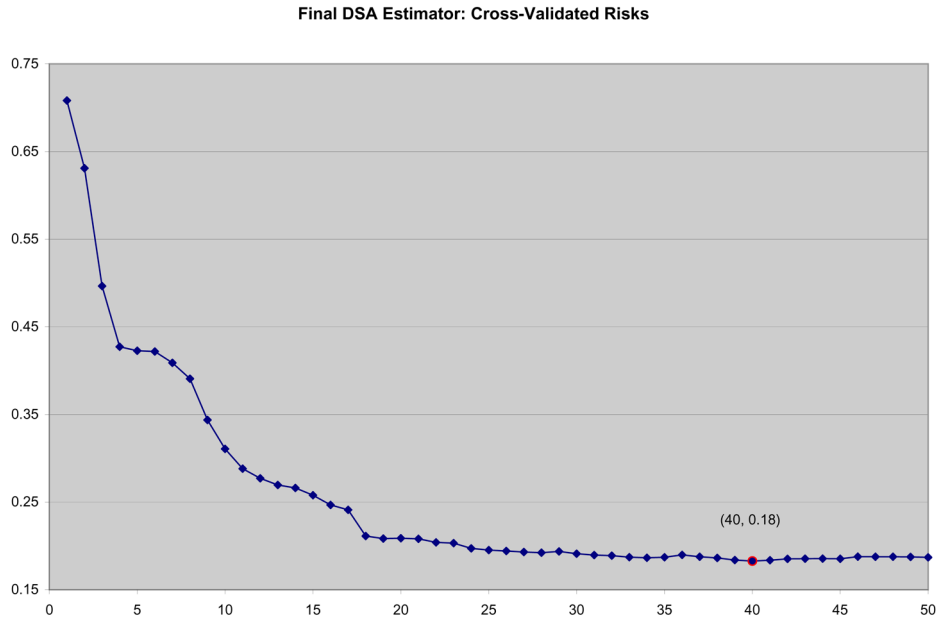

Figure 3.
D/S/A Estimator applied to learning sample, sizes ∈ {1,… ,50}

Table 1

Simulated Example

Super Learner: Cross-Validated Risks of Candidate Learners (n=500). "Linear Regression/LARS (1)" refers to a Linear Regression/LARS fit with main terms only, "Linear Regression/LARS (2)" refers to a Linear Regression/LARS fit with main terms and all 2-way interactions.

Method	Median	Mean	Std Error
Linear Regression (1)	4.477	4.414	0.76
Linear Regression (2)	1.182	1.165	0.16
LARS (1)	4.594	4.719	0.92
LARS (2)	1.179	1.183	0.13
Logic Regression	1.026	1.043	0.21
D/S/A	1.026	1.055	0.19
CART	1.773	1.828	0.60
Ridge Regression	1.176	1.157	0.16

Table 2

Simulated Example

Super Learner: Comparing Logic Regression and the D/S/A Algorithm. Shows the fine-tuning parameters selected (number of leaves for Logic Regression, number of terms for D/S/A) and the associated cross-validated risks across the 10 partitions of the learning set into training and validation sets. Note: The additional fine-tuning parameters were selected consistently across the 10 training sets (cross-validation always selected 5 trees for Logic Regression and 2-way interactions for D/S/A).

	Logic Regression		D/S/A Algorithm	
Sample	**Leaves**	**CV Risk**	**Terms**	**CV Risk**
1	11	1.207	5	1.021
2	19	0.811	5	1.084
3	10	0.988	5	1.066
4	17	1.450	5	0.996
5	14	1.028	5	1.064
6	14	0.812	6	1.057
7	18	0.810	5	1.064
8	11	1.193	5	1.034
9	11	1.110	5	1.031
10	18	1.025	5	1.061
ave	14.3	1.043	5.1	1.055

Table 3

Simulated Example

Super Learner: Cross-Validated Risks of Candidate Learners (n=100). "linear regression/LARS (1)" refers to a least squares/LARS fit with main terms only, "linear regression/LARS (2)" refers to a linear regression/LARS fit with main terms and all 2-way interactions.

Method	Median	Mean	Std Error
Linear Regression (1)	4.311	4.853	1.93
Linear Regression (2)	1.537	2.256	1.57
LARS (1)	5.438	5.755	2.72
LARS (2)	1.539	2.111	1.41
Logic Regression	1.351	1.285	0.41
D/S/A	0.838	0.982	0.38
CART	2.852	3.795	1.84
Ridge Regression	1.312	1.474	0.74

Table 4

Simulated Example

Super Learner: Cross-Validated Risks of Candidate Learners (n=1,000). "linear regression/LARS (1)" refers to a least squares/LARS fit with main terms only, "linear regression/LARS (2)" refers to a linear regression/LARS fit with main terms and all 2-way interactions.

Method	Median	Mean	Std Error
Linear Regression (1)	4.796	4.728	0.67
Linear Regression (2)	1.146	1.134	0.19
LARS (1)	5.015	4.975	0.71
LARS (2)	1.151	1.152	0.20
Logic Regression	1.110	1.058	0.19
D/S/A	1.118	1.066	0.20
CART	1.590	1.528	0.27
Ridge Regression	1.141	1.131	0.18

Table 5

Simulated Example

Super Learner: Cross-Validated Risks of Candidate Learners (n=10,000). "linear regression/LARS (1)" refers to a least squares/LARS fit with main terms only, "linear regression/LARS (2)" refers to a linear regression/LARS fit with main terms and all 2-way interactions.

Method	Median	Mean	Std Error
Linear Regression (1)	4.589	4.588	0.20
Linear Regression (2)	1.032	1.030	0.03
LARS (1)	4.666	4.652	0.23
LARS (2)	1.056	1.047	0.03
Logic Regression	1.027	1.064	0.15
D/S/A	1.027	1.023	0.04
CART	1.333	1.324	0.07
Ridge Regression	1.033	1.029	0.03

Table 6

Simulated Example 2

Super Learner: Cross-Validated Risks of Candidate Learners (n=500).

Method	Median	Mean	Std Error
Logic Regression	2.410	2.368	0.82
D/S/A	1.330	1.426	0.30
Convex	1.239	1.384	0.36

Table 7

Simulated Example 2

Super Learner: True Risks of Candidate Learners (n=500).

Method	Risk
Logic Regression	1.629
D/S/A	1.397
Convex	1.319

Table 8

Nonpolymorphic treatment-selected protease mutations

W	*pr* Mutation	*W*	*pr* Mutation
1	10F	30	55R
2	10R	31	58E
3	11I	32	66F
4	20I	33	67F
5	20T	34	71I
6	20V	35	73A
7	23I	36	73C
8	24I	37	73S
9	30N	38	73T
10	32I	39	74A
11	33F	40	74P
12	34Q	41	74S
13	35G	42	76V
14	43T	43	79A
15	46I	44	82A
16	46L	45	82F
17	46V	46	82S
18	47V	47	82T
19	48M	48	84A
20	48V	49	84C
21	50L	50	84V
22	50V	51	85V
23	53L	52	88D
24	54A	53	88S
25	54L	54	88T
26	54M	55	89V
27	54S	56	90M
28	54T	57	92R
29	54V	58	95F

Table 9

Super Learner: Cross-Validated Risks

Method	Median	Mean	Std Error
Linear Regression	0.175	0.187	0.039
LARS	0.192	0.205	0.052
Logic Regression	0.219	0.256	0.115
D/S/A	0.174	0.188	0.041
CART	0.273	0.335	0.228
Ridge Regression	0.199	0.208	0.058

Table 10

Linear Regression Model (Significance codes: 0 = ***, 0.001 = **, 0.01 = *, 0.05 = .)

Term	β	SE	*t*-stat	*p*-value
(Intercept)	−1.00710	0.02264	−44.488	< 2e-16 ***
10F	0.17563	0.05588	3.143	0.001743 **
10R	0.04947	0.20014	0.247	0.804860
11I	−0.13140	0.14178	−0.927	0.354371
20I	0.48555	0.06452	7.525	1.67e-13 ***
20T	0.18773	0.09784	1.919	0.055438 .
20V	0.06381	0.22119	0.288	0.773056
23I	0.11457	0.11941	0.959	0.337687
24I	0.41711	0.08131	5.130	3.78e-07 ***
30N	1.25778	0.08336	15.088	< 2e-16 ***
32I	0.26527	0.10747	2.468	0.013818 *
33F	−0.14549	0.07144	−2.036	0.042099 *
34Q	0.10543	0.26001	0.405	0.685243
35G	−0.27326	0.27505	−0.993	0.320820
43T	0.23510	0.09581	2.454	0.014384 *
46I	0.29382	0.04603	6.383	3.21e-10 ***
46L	0.10619	0.06182	1.718	0.086304 .
46V	−0.01298	0.23730	−0.055	0.956410
47V	−0.08614	0.13087	−0.658	0.510622
48M	0.53284	0.26823	1.986	0.047379 *
48V	0.22471	0.09295	2.418	0.015886 *
50L	−0.84575	0.11336	−7.461	2.63e-13 ***
50V	0.13139	0.11513	1.141	0.254199
53L	0.04208	0.08835	0.476	0.633978
54A	1.64700	0.39027	4.220	2.77e-05 ***
54L	0.21353	0.10413	2.051	0.040689 *
54M	0.44536	0.13024	3.419	0.000665 ***
54S	1.46285	0.32910	4.445	1.03e-05 ***
54T	1.75388	0.22332	7.854	1.57e-14 ***
54V	0.56756	0.05273	10.764	< 2e-16 ***

Table 11

Linear Regression Model (cont'd)

Term	$\hat{\beta}$	SE	*t*-stat	*p*-value
55R	0.21188	0.10159	2.086	0.037384 *
58E	0.21815	0.08184	2.665	0.007870 **
66F	0.24775	0.17450	1.420	0.156138
67F	0.66268	0.24346	2.722	0.006656 **
71I	0.07484	0.11650	0.642	0.520804
73A	0.03265	0.21218	0.154	0.877737
73C	0.14335	0.15814	0.906	0.365016
73S	0.44710	0.06210	7.199	1.60e-12 ***
73T	0.46391	0.10172	4.560	6.05e-06 ***
74A	0.05345	0.39476	0.135	0.892345
74P	0.53279	0.15491	3.439	0.000618 ***
74S	0.45321	0.09666	4.689	3.32e-06 ***
76V	−0.09230	0.08718	−1.059	0.290075
79A	0.73175	0.41289	1.772	0.076799 .
82A	0.30910	0.05866	5.269	1.84e-07 ***
82F	0.61187	0.11130	5.497	5.45e-08 ***
82S	0.42036	0.29461	1.427	0.154085
82T	0.25881	0.07793	3.321	0.000945 ***
84A	2.17172	0.15347	14.151	< 2e-16 ***
84C	1.76901	0.14486	12.212	< 2e-16 ***
84V	0.31758	0.04599	6.906	1.15e-11 ***
85V	−0.21926	0.09819	−2.233	0.025868 *
88D	0.42180	0.08864	4.758	2.38e-06 ***
88S	1.09265	0.07317	14.933	< 2e-16 ***
88T	0.55475	0.40114	1.383	0.167139
89V	0.05987	0.15417	0.388	0.697893
90M	0.64667	0.04185	15.453	< 2e-16 ***
92R	0.04901	0.43799	0.112	0.910932
95F	0.31722	0.20472	1.550	0.121722

Table 12

D/S/A Estimator: *Best* Model of Sizes 1 to 20. (i.e., Best model of size 1: L90M, Best Model of Size 2: L90M and 30N, etc.)

Mutation	Order	Mutation	Order
90M	1	20I	11
30N	2	50L	12
54V	3	73S	13
46I	4	24I	14
84C	5	54S	15
84A	6	74S	16
88S	7	82F	17
54T	8	10F	18
84V	9	54M	19
82A	10	88D	20