University of California, Berkeley

U.C. Berkeley Division of Biostatistics Working Paper Series

Year 2009 *Paper* 244

Selecting Optimal Treatments Based on Predictive Factors

Eric C. Polley[*] Mark J. van der Laan[†]

[*]University of California, Berkeley, eric.polley@nih.gov

[†]University of California - Berkeley, laan@berkeley.edu

University of California, Berkeley

U.C. Berkeley Division of Biostatistics Working Paper Series

Year 2009 *Paper* 1

Selecting Optimal Treatments Based on Predictive Factors

Eric C. Polley* Mark J. van der Laan†

*University of California, Berkeley, ecpolley@berkeley.edu

†University of California - Berkeley, laan@berkeley.edu

19.1 Introduction

With the increasing interest in individualized medicine there is a greater need for robust statistical methods for prediction of optimal treatment based on the patient's characteristics. When evaluating two treatments, one treatment may not be uniformly superior to the other treatment for all patients. A patient characteristic may interact with one of the treatments and change the effect of the treatment on the response. Clinical trials are also collecting more information on the patient. This additional information on the patients combined with the state-of-the-art in model selection allows researchers to build better optimal treatment algorithms.

In this chapter we introduce a methodology for predicting optimal treatment. The methodology is demonstrated first on a simulation and then on a phase III clinical trial in neuro-oncology.

19.2 Predicting Optimal Treatment Based on Baseline Factors

Start with a randomized controlled trial where patients are assigned to one of two treatment arms, $A \in \{0, 1\}$, with $\Pr(A = 1) = \Pi_A$. The main outcome for the trial is defined at a given time point t as $Y = \mathrm{I}(T > t)$ where T is the survival time. For example, the main outcome may be the six-month progression-free rate and T is the progression time. Also collected at the beginning of the trial is a set of baseline covariates W. The baseline covariates may be any combination of continuous and categorical variables. The baseline covariates can be split into prognostic and predictive factors. Prognostic factors are patient characteristics which are associated with the outcome independent of the treatment given, while predictive factors are patient characteristics which interact with the treatment in their association with the outcome. To determine the optimal treatment, a model for how the predictive factors and treatment are related to the outcome needs to be estimated.

The observed data is $O_i = (W_i, A_i, Y_i = \mathrm{I}(T_i > t)) \sim P$ for $i = 1, \ldots, n$. For now assume Y is observed for all patients in the trial but this assumption is relaxed in the next section.

The optimal treatment given a set of baseline variables is found using the W-specific variable importance parameter:

$$\Psi(W) = \mathrm{E}(Y|A = 1, W) - \mathrm{E}(Y|A = 0, W) \tag{19.1}$$

$\Psi(W)$ is the additive risk difference of treatment A for a specific level of the prognostic variables W. The conditional distribution of Y given W is defined as $\{Y|W\} \sim \mathrm{Bernoulli}(\pi_Y)$. The subscript W is assumed on π_Y and left off for clarity of the notation. Adding the treatment variable A into the conditioning statement we define $\{Y|A = 1, W\} \sim \mathrm{Bernoulli}(\pi_{+1})$ and $\{Y|A = 0, W\} \sim \mathrm{Bernoulli}(\pi_{-1})$. Again the subscript W is dropped for clarity but assumed throughout the paper. The parameter of interest can be expressed as $\Psi(W) = \pi_{+1} - \pi_{-1}$. For a given value of W, $\Psi(W)$ will fall into one of three intervals with each interval leading to a different treatment decision. The three intervals for $\Psi(W)$ are:

1. $\Psi(W) > 0$: indicating a beneficial effect of the intervention $A = 1$.
2. $\Psi(W) = 0$: indicating no effect of the intervention A.
3. $\Psi(W) < 0$: indicating a harmful effect of the intervention $A = 1$.

Knowledge of $\Psi(W)$ directly relates to knowledge of the optimal treatment.

As noted in [1], the parameter of interest can be expressed as:

$$\Psi(W) = \mathrm{E}\left(\left(\frac{\mathrm{I}(A = 1)}{\Pi_A} - \frac{\mathrm{I}(A = 0)}{1 - \Pi_A}\right) Y|W\right). \tag{19.2}$$

When $\Pi_A = 0.5$, the conditional expectation in equation (19.2) can be modeled with the regression of $Y(A - (1 - A))$ on W. Let $Z = Y(A - (1 - A))$ and since A and Y are binary variables:

$$Z = \begin{cases} +1 & \text{if } Y = 1 \;\&\; A = 1 \\ 0 & \text{if } Y = 0 \\ -1 & \text{if } Y = 1 \;\&\; A = 0 \end{cases}$$

The observed values of Z follow a multinomial distribution. The parameter $\Psi(W)$ will be high dimensional in most settings and the components of $\Psi(W)$ are effect modifications between W and the treatment A on the response Y. The parameter can be estimated with a model $\Psi(W) = m(W|\beta)$. The functional form of $m(W|\beta)$ can be specified *a priori*, but since the components of the model represent effect modifications, knowledge of a reasonable model may not be available and we recommend a flexible approach called the super learner (described in the next section) for estimating $\Psi(W)$. In many cases a simple linear model may work well for $m(W|\beta)$, but as the true functional form of $\Psi(W)$ becomes more complex, the super learner gives the researcher flexibility in modeling the optimal treatment function. With the squared error loss function for a specific model $m(W|\beta)$, the parameter estimates are:

$$\beta_n = \arg\min_{\beta} \sum_{i=1}^{n} \left(Z_i - m(W_i|\beta)\right)^2 \tag{19.3}$$

The treatment decision for a new individual with covariates $W = w$ is to treat with $A = 1$ if $m(w|\beta_n) > 0$, otherwise treat with $A = 0$.

A normal super learner model for $m(W|\beta)$ would allow for a flexible relationship between W and Z but these models do not respect the fact that $\Psi(W)$ is bounded between -1 and $+1$. The regression of Z on W does not use the information that the parameter $\Psi(W) = \pi_{+1} - \pi_{-1}$ is bounded between -1 and $+1$. The estimates in equation (19.3) have a nice interpretation since the model predicts the additive difference in survival probabilities. In proposing an alternative method, we wanted to retain the interpretation of an additive effect measure but incorporate the constrains on the distributions. Starting with the parameter of interest in equation (19.1) we add a scaling value based on the conditional distribution of Y given W as in:

$$\Psi'(W) = \frac{\mathrm{E}_P(Y|A=1,W) - \mathrm{E}_P(Y|A=0,W)}{\mathrm{E}_P(Y|W)} = \frac{\pi_{+1} - \pi_{-1}}{\pi_Y} \tag{19.4}$$

Since $\pi_Y = \Pr(Y = 1|W) = \Pr(Z \neq 0|W)$, the new parameter $\Psi'(W) = \mathrm{E}(Z|Z \neq 0, W)$. When we restrict the data to the cases with $Z \neq 0$ (i.e. $Y = 1$) the outcome becomes a binary variable and binary regression methods can be implemented. For example, the logistic regression model:

$$\mathrm{logit}\left(\Pr(Z = 1|Z \neq 0, W)\right) = m'(W_i|\beta) \tag{19.5}$$

The treatment decision is based on $m'(W_i|\beta_n) > 0$ where β_n is the maximum likelihood estimate for the logistic regression model. With the binary regression setting, we are now incorporating the distribution information in creating the prediction model, but losing information by working on a subset of the data. These trade-offs depend on the probability π_Y and we will evaluate both methods on the trial example below. In the next section we propose a data-adaptive method for estimating $\Psi(W)$.

19.3 Super Learner

Many methods exist for prediction, but for any given data set it is not known which method will give the best prediction. A good prediction algorithm should be flexible to the true data generating distribution. One such algorithm is the super learner [2]. The super learner is applied to predict the optimal treatment based on the observed data. The super learner algorithm starts with the researcher selecting a set of candidate prediction algorithms (candidate learners). This list of candidate learners should be selected to cover a wide range of basis functions. The candidate learners are selected prior to analyzing the data; selection of the candidates based on performance on the observed data may introduce bias in the final prediction model. A flow diagram for the super learner algorithm is provided in figure 19.1. With the candidate learners selected and the data collected, the initial step is to fit all of the candidate learners on the entire data set and save the predicted values for $\Psi_n(W) = m_j(W|\beta_n)$, where j indexes the candidate learners. The data is then split into V equal sized and mutually exclusive sets as is typically done for V-fold cross-validation. Patients in the v^{th} fold are referred to as the v^{th} validation set, and all patients not in the v^{th} fold are referred to as the v^{th} training set. For the v^{th} fold, each candidate learner is fit on the patients in the v^{th} training set and the predicted values for $\Psi(W) = m_j(W|\beta_n)$ for the patients in the v^{th} validation set are saved. This process of training the candidate learners on the out of fold samples and saving the predicted values in the fold is repeated for all V folds. The predictions from all V folds are stacked together in a new data matrix X^v. With the prediction data, regress the observed outcome Z on the columns of X^v, which represent the predicted outcomes for each candidate learner. This regression step selects weights for each candidate learner to minimize the cross-validated risk. With the estimates, β_n, from the model $\mathrm{E}(Z|X^v) = m(X|\beta)$ the super learner only saves the weights (β_n) and the functional form of the model. The super learner prediction is then based on combining the predictions from each candidate learner on the entire data set with the weights from the cross-validation step.

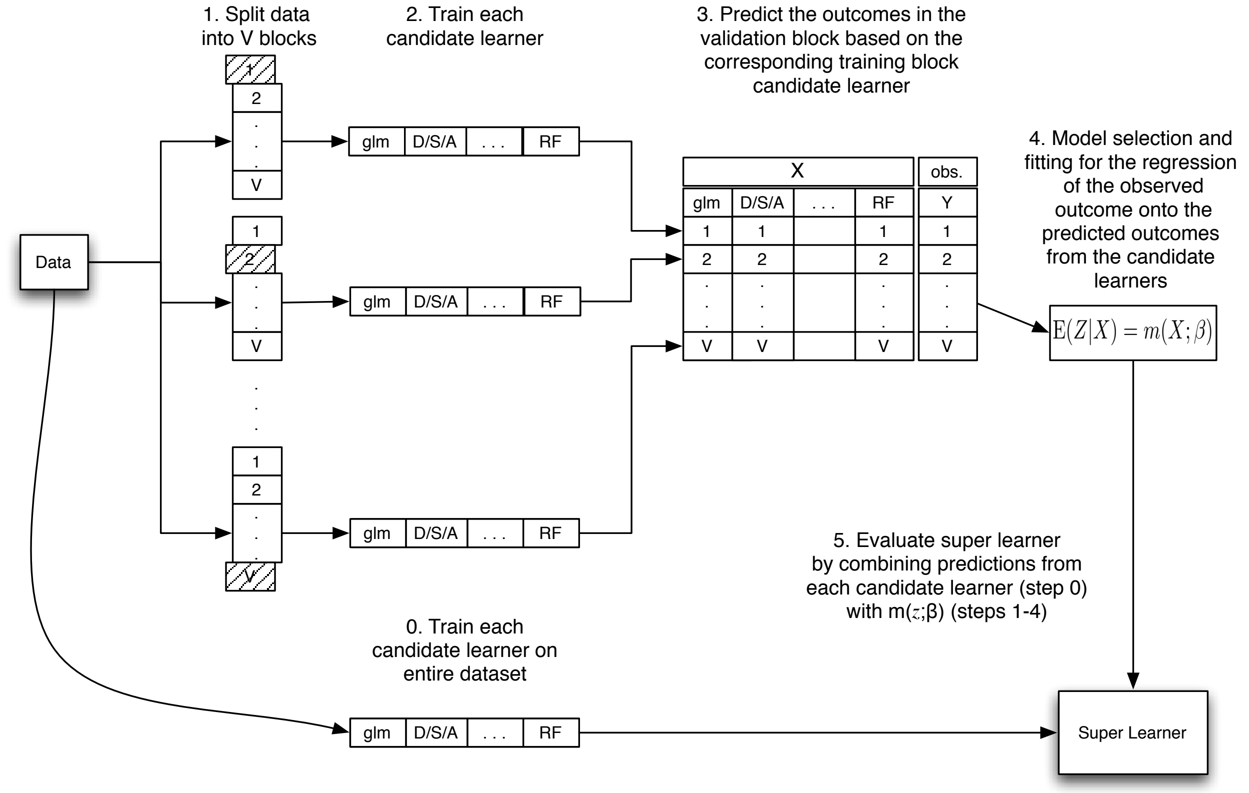
1. Split data into V blocks
2. Train each candidate learner
3. Predict the outcomes in the validation block based on the corresponding training block candidate learner
4. Model selection and fitting for the regression of the observed outcome onto the predicted outcomes from the candidate learners
5. Evaluate super learner by combining predictions from each candidate learner (step 0) with m(z;β) (steps 1-4)
0. Train each candidate learner on entire dataset
Data
1
2
V
glm
D/S/A
. . .
RF
X
obs.
Y
E(Z|X) = m(X;β)
Super Learner

Figure 19.1: Flow diagram for super learner

19.4 Extensions for Censored Data

In a prospective trial the data may be subject to right censoring. In both methods above, right censoring leads to the outcome Z being missing. The data structure is extended to include an indicator for observing the outcome. Let C be the censoring time (for individuals with an observed outcome we set $C = \infty$). Define $\Delta = \mathrm{I}(C > t)$. $\Delta = 1$ when the outcome is observed and $\Delta = 0$ when the outcome is missing. The observed data is the set $(W, A, \Delta, Y\Delta)$. For the first method, we propose using the doubly robust censoring unbiased transformation [3]. The doubly robust censoring unbiased transformation generates a new variable Z^* which is a function of the observed data but has the additional property:

$$\mathrm{E}\left(Z^*|W, \Delta = 1\right) = \mathrm{E}\left(Z|W\right)$$

The transformation allows estimation of the parameter $\Psi(W)$ by applying the super learner on the uncensored observations with the transformed variable Z^* as the outcome. The doubly robust censoring unbiased transformation is:

$$Z^* = \frac{Z\Delta}{\pi(W)} - \frac{\Delta}{\pi(W)} Q(W) + Q(W), \tag{19.6}$$

where $\pi(W) = \Pr(\Delta = 1|W)$ and $Q(W) = \mathrm{E}(Z|W, \Delta = 1)$. Both $\pi(W)$ and $Q(W)$ need to be estimated from the data. If either $\pi(W)$ or $Q(W)$ is consistently estimated, then the prediction function $\mathrm{E}(Z^*|W, \Delta = 1) = m(W|\beta_n)$ is an unbiased estimate for the true parameter $\Psi(W)$. The censoring mechanism $\pi(W)$ can be estimated with a logistic regression model or a binary super learner on the entire data set. Similarly, $Q(W)$ may be fit with a linear regression model or a super learner, but on the subset of the data with observed values for Z.

For the second method which relies on modeling $\mathrm{E}(Z|Z \neq 0, W)$, the main feature was the ability to use the knowledge of the distributions to develop a better model. To retain the binary outcome, the doubly robust censoring unbiased transformation will not work. An alternative method for the right censoring which will retain the binary outcome would be inverse probability of censoring weighting. Inverse probability of censoring weights uses the same $\pi(W)$ as above, but does not incorporate the other nuisance parameter $Q(W)$. When applying the binary super learner for $\mathrm{E}(Z|Z \neq 0, W, \Delta = 1)$ the weights $1/\pi(W)$ will be applied for both the candidate learners and the V-fold cross-validation steps. The super learner will minimize the weighted loss function.

19.5 Simulation Example

We first demonstrate the proposed method on a simulation example where the true value of $\Psi(W)$ is known. The baseline variables were all simulated as normally distributed, $W_j \sim$

$N(0,1)$, $j = 1, \dots, 10$. The treatment was randomly assigned with $\Pi_A = 0.5$. The true model for the outcome was:

$$\begin{aligned}\Pr(Y=1|A,W) = g^{-1}(&0.405A - 0.105W_1 + 0.182W_2 + 0.039AW_2 \\ &+0.006AW_2W_3 - 0.357AW_4 - 0.020AW_5W_6 - 0.051AW_6)\end{aligned} \tag{19.7}$$

Where $g^{-1}(\cdot)$ is the inverse logit function and W_j refers to the j^{th} variable in W. The true model was selected to include interactions between the treatment and some of the baseline variables. With knowledge of the true model for the outcome Y, the true value of $\Psi(W)$ is calculated for every individual.

The first method involves the regression of Z on W. We applied the super learner for $m(W|\beta)$. 10-fold cross validation was used for estimating the candidate learner weights in the super learner. The super learner for the first method included five candidate learners. The first candidate was ridge regression [4]. Ridge regression used an internal cross validation to select the penalty parameter. Internal cross validation means the candidate learner performed a V-fold cross validation procedure within the folds for the super learner. Structurally, when the candidate learner also performs cross validation within the super learner cross validation we have nested cross validation; therefore, we refer to the candidate learner cross validation as internal cross validation. The second candidate was random forests [5]. For the random forest candidate learner, 1000 regression trees were grown. The third candidate was least angle regression [6]. An internal 10-fold cross validation procedure was used to determine the optimal ratio of the L1 norm of the coefficient vector compared to the L1 norm of the full least squares coefficient vector. The fourth candidate was adaptive regression splines for a continuous outcome [7]. The final candidate was linear regression. Table 19.1 contains reference for the R packages implemented for the candidate learners in the super learner.

The prediction model from the super learner is:

$$\Psi_n(W) = -0.01 + 7.24(X_n^{ridge}) + 1.16(X_n^{rf}) - 0.20(X_n^{lars}) - 7.07(X_n^{lm}) - 0.03(X_n^{mars})$$

Where X_n^j is the predicted value for Z based on the j^{th} candidate learner. $j = ridge$ is the

Method	R Package	Authors
Adaptive Regression Splines	polspline	Kooperberg
Least Angle Regression	lars	Efron and Hastie
Penalized Logistic	stepPlr	Park and Hastie
Random Forests	randomForest	Liaw and Wiener
Ridge Regression	MASS	Venables and Ripley

Table 19.1: R Packages for Candidate Learners. R is available at `http://www.r-project.org`

ridge regression model. $j = rf$ is the random forests model. $j = lars$ is the least angle regression model. $j = lm$ is the main effects linear regression model. $j = mars$ is the adaptive regression splines model. The largest weights are for ridge regression and the linear regression model. For example, the estimates for the linear regression model is:

$$\begin{aligned} X_n^{lm} = 0.06 &+ 0.02W_1 + 0.01W_2 - 0.03W_3 - 0.07W_4 + 0.01W_5 + 0.05W_6 \\ &- 0.02W_7 - 0.00W_8 - 0.01W_9 - 0.06W_{10}. \end{aligned}$$

The linear regression model has the largest coefficient on W_4, which is the variable with the strongest effect modification with the treatment in the true model (equation (19.7)). The second largest coefficient is on W_{10} which is a variable unrelated to the outcome. The super learner helps smooth over these errors by having multiple candidate learners. For example, W_{10} has a small coefficient (-0.01) in the ridge regression model. When all the candidates are combined into the final super learner prediction model the spurious effect estimates will often disappear resulting in a better predictor. The third largest coefficient from the linear regression model is on W_6 which is also a strong effect modifier in the true model. To evaluate how the super learner is performing in comparison to the other candidate learners, each candidate learner was also fit as a separate estimate. We looked at two risk values, first the $\mathrm{E}(\Psi_n(W) - Z)^2$ which was minimized by each algorithm. For the simulation, the risk $\hat{E}(Z - \Psi(W))^2 = 0.540$ gives a lower bound for the risk $\mathrm{E}(\Psi_n(W) - Z)^2$. Since the true $\Psi(W)$ is known in the simulation, the risk $\mathrm{E}(\Psi_n(W) - \Psi(W))^2$ was also evaluated. Table 19.2 contains the risk values for the simulation. The super learner achieved the smallest $\mathrm{E}(\Psi_n(W) - Z)^2$ and is comparable to MARS and LARS on the risk for the true parameter value $\Psi(W)$.

	$\mathrm{E}(\Psi_n(W) - \Psi(W))^2$	$\mathrm{E}(\Psi_n(W) - Z)^2$
Super Learner	0.012	0.544
MARS	0.012	0.549
LARS	0.012	0.549
Ridge	0.026	0.558
Linear Model	0.028	0.559
Random Forests	0.038	0.565

Table 19.2: Risk for all candidate learners and the super learner

The super learner for the second method included three candidate learners. The first candidate was adaptive regression splines for polychotomous outcomes [8]. The second candidate was the step-wise penalized logistic regression algorithm [9]. The final candidate was main terms logistic regression. The super learner for the second method is:

$$\Psi'_n(W) = -1.20 + 1.43(X_n^{poly}) - 0.50(X_n^{plr}) + 1.61(X_n^{glm})$$

Where X_n^j is the predicted value for Z based on the j^{th} candidate learner. $j = poly$ is the polyclass adaptive spline model. $j = plr$ is the penalized logistic regression model. $j = glm$ is the main effects logistic regression model.

19.6 Example of Prediction Model on Clinical Trial

A phase III clinical trial was conducted to evaluate a novel treatment for brain metastasis. The study recruited 554 patients with newly diagnosed brain metastasis and the patients were randomized to receive either standard care ($A = 0$) or the novel treatment ($A = 1$). The researchers were interested in determining an optimal treatment to maximize the probability of surviving 6 months from treatment initiation without progression. Of the 554 patients, 246 are censored prior to 6 months. For the 308 patients with an observed 6 month progression time, 130 progressed or died (42.2%). In addition to the treatment and event time data, the researchers collected baseline prognostic and predictive factors on every patient. We apply the super learner to estimate a model for selecting the optimal treatment given a patient's baseline factors. A breakdown of the sample size and treatment allocations available for each method is given in table 19.3.

		A	
	total	0	1
Enrolled	554	275	279
Method 1	308	158	150
Method 2	130	67	63

Table 19.3: Number of subjects in each treatment arm at enrollment and available for each method.

19.6.1 Super learner for optimal treatment decisions

Both methods proposed above were applied to the data. The first method looks for a model of Z on W treating Z as a continuous variable. The second method looks for a model of Z on W conditional on $Z \neq 0$ treating the outcome as binary.

The same super learners from the simulation example above were used here in the trial example. The predicted model for the first method is:

$$\Psi_n(W) = -0.01 + 0.02(X_n^{ridge}) + 1.21(X_n^{rf}) - 0.84(X_n^{lars}) - 0.28(X_n^{lm}) + 0.50(X_n^{mars})$$

Where X_n^j is the predicted value for Z based on the j^{th} candidate learner. $j = ridge$ is the ridge regression model. $j = rf$ is the random forests model. $j = lars$ is the least angle regression model. $j = lm$ is the main effects linear regression model. $j = mars$ is the adaptive regression splines model. The coefficient estimates for each candidate learner from the super learner can be interpreted as a weight for each candidate learner in the final prediction model. Random forests has the largest absolute weight. When interpreting the weights, be cautious of the often near collinearity of the columns of X. To evaluate the super learner in comparison to the candidate learners, a 10-fold cross validation of the super learner and each of the candidate learners themselves was used to estimate $\mathrm{E}(\Psi_n(W) - Z)^2$. Table 19.4 contains the risk estimates. For the trial example, both the lars algorithm and the mars algorithm outperform the super learner. As observed in the simulation, minimizing the risk $\mathrm{E}(\Psi_n(W) - Z)^2$ should directly relate to minimizing the risk $\mathrm{E}(\Psi_n(W) - \Psi(W))^2$. These cross-validation estimates may be used to select an optimal final model for the treatment decisions.

The second method evaluates $\mathrm{E}(Z|Z \neq 0, W) = m'(W|\beta)$. The estimated super learner

Method	Risk
Lars	0.426
Mars	0.426
Super Learner	0.445
Ridge Regression	0.505
Random Forests	0.509
Linear Model	0.525

Table 19.4: 10-fold honest cross validation estimates of $\mathrm{E}(\Psi_n(W) - Z)^2$ for the super learner and each of the candidate learners on their own.

model for the second method is:

$$\Psi_n'(W) = -0.53 - 0.40(X_n^{poly}) + 0.55(X_n^{plr}) + 0.81(X_n^{glm})$$

Where X_n^j is the predicted value for Z based on the j^{th} candidate learner. $j = poly$ is the polyclass adaptive spline model. $j = plr$ is the penalized logistic regression model. $j = glm$ is the main effects logistic regression model. To compare the two methods, we created a confidence interval at the mean vector for W. Let $\bar{w}$ be the vector of observed means for the baseline variables using all observations in the trial. Confidence intervals were

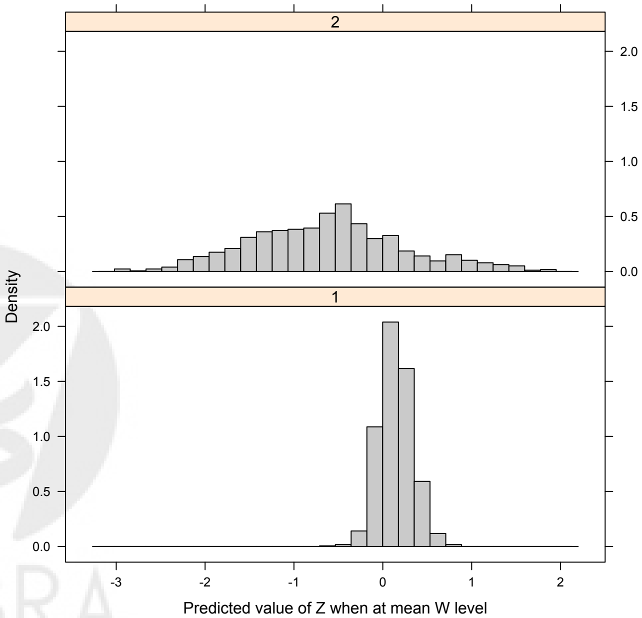

Figure 19.2: Histograms from 1000 bootstrap samples for $\Psi'(W = \bar{w})$ and $\Psi(W = \bar{w})$. The number in the title bar refers to the method used.

created based on 1000 bootstrap samples of the entire super learner. The 95% confidence interval for $m(\bar{w}|\beta)$ based on the first method is $(-0.20, 0.52)$. The 95% for $m(\bar{w}|\beta)$ based on the second method is $(-2.23, 1.23)$. Although the second method is able to use the distributional information, the penalty for the smaller sample size is great (308 patients for the first method down to 130 patients for the second method). As can be seen in figure 19.2, the second method has a wide confidence interval compared to the first method.

19.7 Variable Importance Measure

An additional feature of having a good prediction model is better variable importance measures. The variables in $\mathrm{E}(Z|W)$ are effect modifications and when applying the targeted maximum likelihood estimation (tMLE) variable importance measure [10] the results will be causal effect modification importance measures. The targeted maximum likelihood effect modification variable importance allows the researcher to focus on each variable in W individually while adjust for the other variables in W. An initial variable importance estimate is based on an univariate regression, $Z^* = \beta_{0j} + \beta_{1j} W_j,\ j = 1, \ldots, p$ where p is the number of baseline covariates in W. The top 5 baseline variables based on the ranks of the univariate p-values is presented in table 19.5. The top unadjusted effect modification variable is an indicator of whether the patients lives in the US or Europe, followed by an indicator for the patients being in RPA class 2, an indicator for the primary tumor being controlled, an indicator for extracranial metastasis, and finally an indicator for the patient's age greater than 65 years. The top 5 baseline variables from the LARS procedure are similar to those from the univariate regression with the exception of Squamous cell indicator replacing the RPA class 2 indicator. For the tMLE variable importance, the effect of W_j on Z is adjusted by all other covariates in W. Let $W_{(-j)}$ be all covariates in W excluding the j^{th} variable. The targeted maximum likelihood variable importance measure as outlined in [11] was then applied using the predictions from the super learner as the initial estimate of $\mathrm{E}(Z|W)$. The targeted effect modification parameter is then:

$$\psi_j = \mathrm{E}\left(\mathrm{E}(Z|W_j = 1, W_{(-j)}) - \mathrm{E}(Z|W_j = 0, W_{(-j)})\right),\ j = 1, \ldots, p \qquad (19.8)$$

Method	Baseline Variable	Effect	p-value
Univariate Regression	US vs Europe	-0.222	0.007
	RPA class 2	-0.229	0.017
	Primary tumor control	0.165	0.052
	Extracranial mets	-0.133	0.069
	Age > 65 years	-0.157	0.075
LARS	US vs Europe	-0.124	0.350
	Primary tumor control	0.080	0.405
	Age > 65 years	-0.050	0.412
	Extracranial mets	-0.028	0.413
	Squamous cell	0.034	0.419
tMLE	Mets Dx > 6 Mo	0.864	<0.001
	Squamous cell	1.012	<0.001
	Adeno carcinoma	0.129	0.007
	Extracranial mets	-0.102	0.022
	Caucasian	0.172	0.035

Table 19.5: Top 5 effect modifiers based on univariate regression, lars, and super learner with targeted maximum likelihood. The standard error was based on a bootstrap with 1,000 samples.

The top 5 baseline variables are presented in table 19.5. The effect estimates from the tMLE procedure can be considered causal effect modifiers. Only extracranial mets appears in both the adjusted and unadjusted top 5 list, although squamous cell indicator does appear in both the LARS procedure and the tMLE procedure. The top variable (Mets Dx $>$ 6 Mo) is an indicator for the metastasis diagnosis occurring greater than 6 months after prvious cancer. The tMLE list contains two indicators for histology of the tumor cells (Squamous and Adeno carcinoma) suggesting that some tumor types my respond better to the treatment compared to others. Comparing the variable importance lists, the indicator for the patient being in the United States compared to Europe is on top of the list for the univariate regression and the lars model, but absent from the tMLE list. There is no biological evidence for geographical location to interact with the treatment in this trial. The variable importance based on targeted maximum likelihood is able to appropriately adjust for the confounding on the other variables in W and remove the US versus Europe indicator from the list of top variables. The variable importance list from the tMLE has a better interpretation and is informative as to which patient characteristics have a causal interaction with the treatment.

19.8 Discussion

Two methods were proposed for predicting the optimal treatment based on baseline factors. The first method involves modeling Z on W disregarding the knowledge that $\mathrm{E}(Z|W)$ is bounded between -1 and $+1$. The second method incorporates the bounds, but does so at a cost in sample size by modeling $\mathrm{E}(Z|Z \neq 0, W)$. The second method predicts a scaled version of the parameter of interest, and so is still valid for making treatment decisions. In the simulation and trial example presented here, the loss of sample size in the second method greatly increased the variability of the final prediction. But both the simulation and trial example had a high fraction of patients with $Z = 0$ (equivalently, $Y = 0$). The second method may outperform the first method in settings where $\Pr(Y = 0)$ is very small. For the examples presented here, no problems were observed with the first method not respecting the bounds on $\mathrm{E}(Z|W)$.

In the trial example, the super learner did perform better than the main terms linear regression based on the estimate of the risk $\mathrm{E}(\Psi_n(W) - Z)^2$. Even though the super learner has shown to have excellent performance across a range of simulations [2, 12] and in various of our data analyses in breast cancer research, there is a risk that the super learner will result in a slight over-fit. In the data analysis we observed that the super learner was ranked third, but competitive with the top two candidate learners, LARS and MARS. We have also proposed an extension to the super learner outlined here to adaptively select the number of candidates [13] so that the weaker candidates are not selected, which we believe will protect the super learner against possible over-fitting, but this was not implemented in the current data analysis yet.

We observed that the difference in sample size between the two methods may make the second method unusable in this example, but the two methods also differed in the treatment of right censoring. The first method incorporated the doubly robust censoring unbiased transformation while the second method used the inverse probability of censoring weights. If the model for $Q(W)$ was correctly specified, but the model for the censoring mechanism was not consistently estimating $\pi(W)$, the doubly robust estimator would still be unbiased but the inverse probability of censoring weighted method will be biased. Alternatively, if $\pi(W)$ was correctly specified, but $Q(W)$ was inconsistent, then both methods will be unbiased.

The doubly robust transformation gives the researcher two chances to correctly the nuisance parameters, while the inverse weighting method relies solely on the model for $\pi(W)$. When there is uncertainty regarding the model for the censoring mechanism, the doubly robust transformation is preferred.

The methods presented above are not limited to randomized clinical trials. Optimal treatment prediction models could also be estimated from observational or registry data sets. As long as the variables needed to estimate $\Pr(A = 1|W))$ are collected in the study the above methods easily extend to the non-randomized setting. Registry data sets are often larger than randomized trials and therefore have more power to detect the interaction effects necessary for predict optimal treatments.

References

[1] M. J. van der Laan. Statistical inference for variable importance. *The International Journal of Biostatistics*, 2, 2006.

[2] M. J. van der Laan, E. C. Polley, and A. E. Hubbard. Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6(25), 2007.

[3] D. Rubin and M. J. van der Laan. Doubly robust censoring unbiased transformations. Technical Report 208, University of California, Berkeley, Division of Biostatistics, 2006.

[4] A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(3):55–67, 1970.

[5] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[6] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[7] J. H. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 19(1):1–141, 1991.

[8] C. Kooperberg, S. Bose, and C. J. Stone. Polychotomous regression. *Journal of the American Statistical Association*, 92:117–127, 1997.

[9] M. Y. Park and T. Hastie. l_1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society, Series B*, 69(4):659–677, 2007.

[10] M. J. van der Laan and D. Rubin. Targeted maximum likelihood learning. *International*

Journal of Biostatistics, 2(1), 2007.

[11] O. Bembom, M. L. Petersen, S. Rhee, W. J. Fessel, S. E. Sinisi, R. W. Shafer, and M. J. van der Laan. Biomarker discovery using targeted maximum likelihood estimation: Application to the treatment of antiretroviral resistant HIV infection. Technical Report 221, University of California, Berkeley, Division of Biostatistics, 2007.

[12] S. E. Sinisi, E. C. Polley, , M. L. Petersen, S.Y. Rhee, and M. J. van der Laan. Super learning: An application to the prediction of HIV-1 drug resistance. *Statistical Applications in Genetics and Molecular Biology*, 6(1), 2007.

[13] E. C. Polley and M. J. van der Laan. Adaptive selection of the functional form for the super learner. in preparation, 2008.