# Targeted Learning: Causal Inference for Observational and Experimental Data

Mark J. van der Laan[1], Maya L. Petersen[1], Sherri Rose[2]

[1]University of California, Berkeley School of Public Health

[2]Johns Hopkins Bloomberg School of Public Health

laan@berkeley.edu · mayaliv@berkeley.edu · srose@jhsph.edu
stat.berkeley.edu/∼laan/
works.bepress.com/maya_petersen/
drsherrirose.com

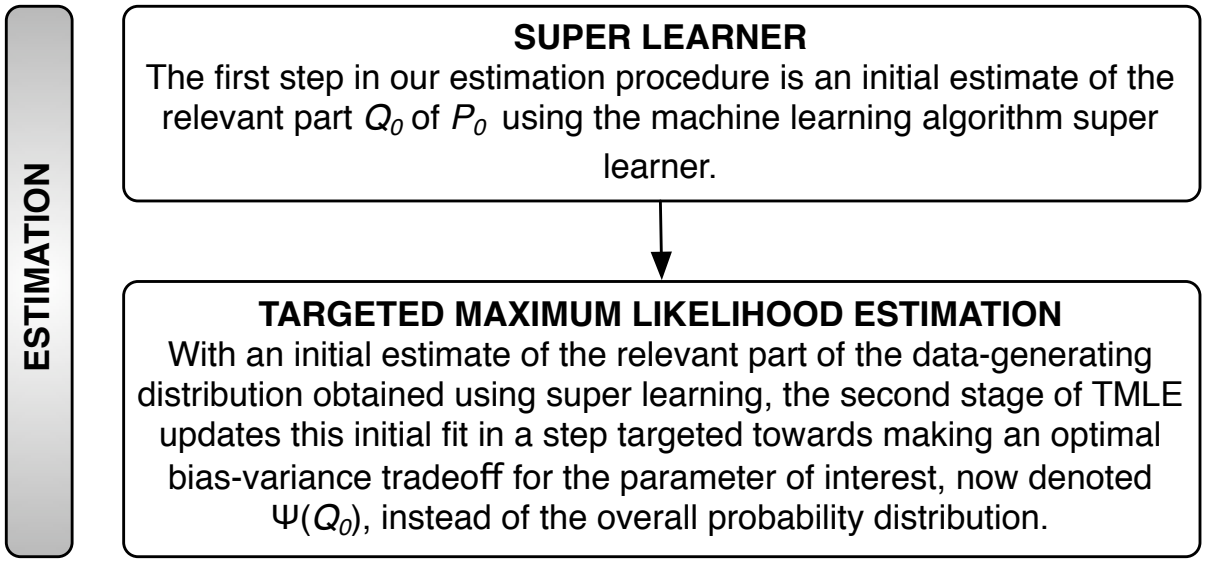
targetedlearningbook.com

July 29, 2012

# LECTURE 3.5

## Super learner

# Road map - Estimation



# Effect Estimation vs. Prediction

Both **effect** and **prediction** research questions are inherently *estimation* questions, but they are distinct in their goals.

**Effect:** Interested in estimating the effect of exposure on outcome adjusted for covariates.

**Prediction:** Interested in generating a function to input covariates and predict a value for the outcome.

An **estimator** is an algorithm that can be applied to any empirical distribution to provide a mapping from the empirical distribution to the parameter space.

# Estimation in Misspecified Parametric Models

In most observational studies, standard practice for prediction and effect estimation involves assuming a parametric statistical model and using maximum likelihood estimation (MLE) to estimate the parameters in that statistical model.

# The goal

What we want is an automated algorithm to semiparametrically estimate $E_0(Y \mid A, W)$ (and $P_0(A = 1 \mid W)$).

In the computer science literature, this is called machine learning.

In statistics these methods are often referred to as data-adaptive.

However, the essential point is that there are semi-parametric methods that also aim to "smooth" the data and estimate this regression function.

# A problem

If the true data-generating distribution is very smooth, a misspecified parametric regression might beat a semiparametric estimator.

This is frustrating!

We want to create a smart semiparametric estimator that is consistent, but in some cases it may "lose" to a misspecified parametric estimator because it is more variable.

# A problem

There are many different potential algorithms we can implement to estimate $E_0(Y \mid A, W)$.

However, how are we to know which one to use a priori?

We cannot bet on a misspecified parametric regression, but we have a problem that one particular algorithm is going to do better than the other candidate estimators.

# The Dangers of Favoritism

- Relative Mean Squared Error (compared to main terms least squares regression) based on the validation sample

| Method | Study 1 | Study 2 | Study 3 | Study 4 |
|---|---|---|---|---|
| Least Squares | 1.00 | 1.00 | 1.00 | 1.00 |
| LARS | 0.91 | 0.95 | 1.00 | 0.91 |
| D/S/A | 0.22 | 0.95 | 1.04 | 0.43 |
| Ridge | 0.96 | 0.9 | 1.02 | 0.98 |
| Random Forest | 0.39 | 0.72 | 1.18 | 0.71 |
| MARS | 0.02 | 0.82 | 0.17 | 0.61 |

# Super Learning in Prediction

| Method | Study 1 | Study 2 | Study 3 | Study 4 | Overall |
|---|---|---|---|---|---|
| Least Squares | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| LARS | 0.91 | 0.95 | 1.00 | 0.91 | 0.95 |
| D/S/A | 0.22 | 0.95 | 1.04 | 0.43 | 0.71 |
| Ridge | 0.96 | 0.9 | 1.02 | 0.98 | 1.00 |
| Random Forest | 0.39 | 0.72 | 1.18 | 0.71 | 0.91 |
| MARS | 0.02 | 0.82 | 0.17 | 0.61 | 0.38 |
| Super Learner | 0.02 | 0.67 | 0.16 | 0.22 | 0.19 |

# SPPARCS Prediction Example

**Background**: Standard practice for mortality risk score prediction relies heavily on regression in parametric statistical models.

**Methods**: Using observational data from the *S*tudy of *P*hysical *P*erformance and *A*ge-*R*elated *C*hanges in *S*onomans (SPPARCS), we generate a function for mortality risk prediction with a machine learning ensembling method.

**Results**: The super learner for predicting death (risk score) outperformed all single algorithms in the collection of algorithms, although its performance was similar to several of the included algorithms.

# Background

Previous studies of elderly populations in the United States have indicated that

- gender,
- smoking status,
- heart health,
- physical activity,
- education level,
- income, and
- weight

are among the important predictors of mortality in elderly populations.

Prediction functions for mortality have been generated in an elderly Northern California population aged 65 and older (Rose et al. 2011) and for nursing home residents with advanced dementia (Mitchell et al. 2010).

# Background

- Standard practice for risk score prediction relies heavily on regression in parametric statistical models, assuming a functional form that is not known.
- Since we typically know very little about how the data were generated, the use of parametric statistical models presents a challenge.

# Background

- Recent medical and epidemiologic studies for prediction have employed newer methods such as random forest and neural networks.
- Researchers are then left with questions such as, *"When should I use random forest instead of standard regression techniques? When should I use neural networks?"*

# Background

Austin et al. Logistic regression had superior performance compared with regression trees for predicting in-hospital mortality in patients hospitalized with heart failure. *J Clin Epidemiol.* 2010; 63(10):1145–55.

Peng et al. Random forest can predict 30-day mortality of spontaneous intracerebral hemorrhage with remarkable discrimination. *Eur J Neurol.* 2010;17(7):945–50.

# Background

- Ensembling methods allow researchers to implement multiple algorithms with an a priori benchmark regarding how to arrive at the final algorithm.
- Investigators therefore do not need to decide beforehand whether to forgo one technique in favor of another; they can use several by incorporating cross-validation.

# Data

- $O = (W, Y)$, where $W$ is a vector of covariates and $Y$ the outcome death.
- Note that $A$ has become one of the $W_i$'s since we are focusing on a prediction question.
- One assumes the actual data as observed in practice can be represented as $n$ i.i.d. observations of the random variable $O \sim P_0$.
- The regression function $E_0(Y \mid W)$, or $P_0(Y = 1 \mid W)$ if $Y$ is binary, is our function of interest for prediction.
- We want the best estimator of this regression function that respects true knowledge about the way the data were generated.

# Data

- The observational cohort data included 2,066 persons aged 54 and over who were residents of Sonoma, CA and surrounding areas in Northern California.
- Enrollment began in May 1993 and concluded in December 1994 with follow-up continuing for approximately 10 years.
- Covariates included self-rated health status, leisure-time physical activity, history of cardiac events, and chronic health conditions at baseline, among others.

# SPPARCS Data Summary

Observational sample (n=2,066) of persons over the age of 54.

- **Outcome** $Y$ was **death** occurring within 5 years of baseline.
- **Covariates** $W = \{W_1, \ldots W_{13}\}$ included self-rated health score and physical activity.

# SPPARCS Data

Table: Selected Characteristics of SPPARCS Data ($n = 2,066$)

| **Variable** | No. | % |
|---|---|---|
| Death ($Y$) | 269 | 13 |
| Female ($W_1$) | 1,225 | 59 |
| Met minimum physical activity level ($W_2$) | 1,460 | 71 |
| Cardiac event prior to baseline ($W_3$) | 356 | 17 |
| Chronic health condition at baseline ($W_4$) | 918 | 44 |
| ... | ... | ... |

# Methods

## Super Learner (van der Laan, Polley, and Hubbard; 2007)

Allows researchers to use multiple algorithms to outperform a single algorithm in non-parametric and semi-parametric statistical models that are based on actual knowledge.

The term algorithm is used very loosely to describe any mapping from the data into a predictor.

# Methods

We define our parameter of interest, $Q_0 = E_0(Y \mid W)$, as the minimizer of the expected squared error loss:

$$Q_0 = \arg\min_Q E_0 L(O, Q),$$

where $L(O, Q) = (Y - Q(A, W))^2$. $E_0 L(O, Q)$, which we want to be small, evaluates the candidate $Q$, and it is minimized at the optimal choice of $Q_0$. We refer to expected loss as the risk.

# Super Learner

## SPPARCS Data: Super Learner

Suppose a researcher is interested in using several different algorithms to estimate $E_0(Y \mid W)$.

We can use these algorithms to build a library of algorithms consisting of all weighted averages of the algorithms.

One of these weighted averages might perform better than one of the algorithms alone. It is this principle that allows us to map a collection of algorithms into a library of weighted averages of these algorithms.

# Methods

It might seem that the implementation of such an estimator is problematic, since it requires **minimizing the cross-validated risk over an infinite set of candidate algorithms** (the weighted averages).

*The contrary is true.*

Super learner is not more computer intensive than the "cross-validation selector" (the single algorithm with the smallest cross-validated risk).

- Only the relatively trivial calculation of the optimal weight vector needs to be completed.

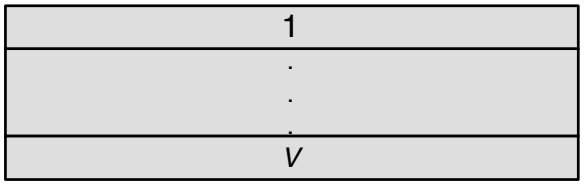
# SPPARCS Data: Super Learner

**1.** Start with the SPPARCS data and a collection of $M$ algorithms. In this analysis $M = 12$.

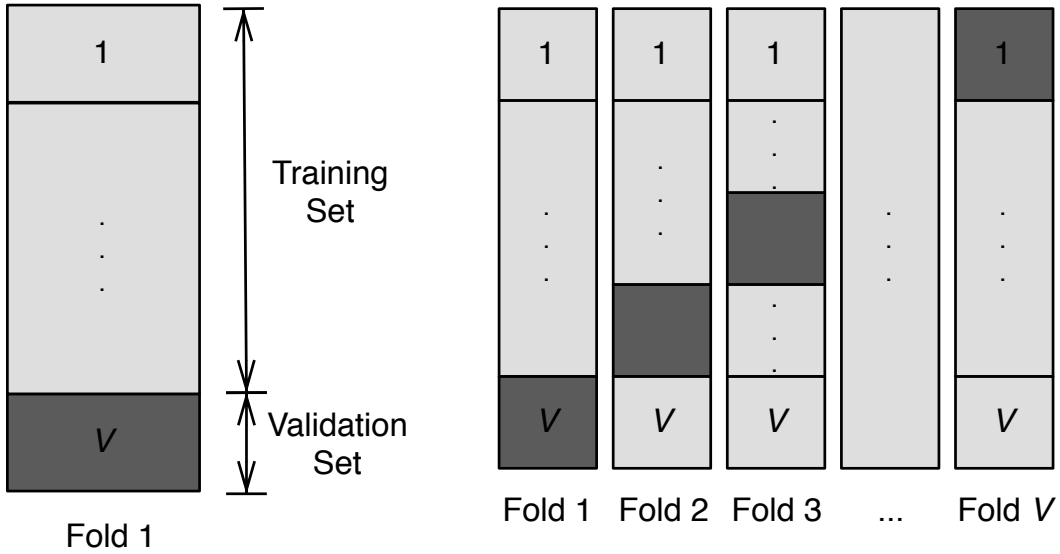
| ID | $W_1$ | ... | $W_{12}$ | $W_{13}$ | Y |
|---|---|---|---|---|---|
| 1 | 1 | ... | 0 | 1 | 1 |
| . | . | . | . | . | . |
| . | . | . | . | . | . |
| . | . | . | . | . | . |
| 2066 | 0 | ... | 1 | 1 | 1 |

bayesglm
glmnet
.
.
.
nnet

**2.** Split the SPPARCS data into $V$ mutually exclusive and exhaustive blocks of equal or approximately equal size. Here $V = 10$.




**3.** Fit each algorithm on the training set for each $V$ fold. For example, in fold 1, our training set could be blocks 1-9, where block 10 will be the validation set. Each algorithm is fit on blocks 1-9. In fold 2, our training set might be blocks 1-8 and block 10 with block 9 serving as the validation set, and so on. At the end of this stage you have $V$ fits for each algorithm.



# SPPARCS Data: Super Learner

**4.** For each algorithm, predict the outcome $Y$ using the validation set in each fold, based on the corresponding training set fit for that fold. At the end of this step you have a vector of predicted values $D_j$, $j=1,\ldots, M$ for each algorithm.

| ID | $D_{bayesglm}$ | ... | $D_{nnet}$ |
|---|---|---|---|
| 1 | 0.54 | ... | 0.42 |
| . | . | . | . |
| . | . | . | . |
| 2066 | 0.09 | ... | 0.12 |

**5.** Compute the estimated CV MSE for each algorithm using the predicted values $D_j$ calculated from the validation sets.

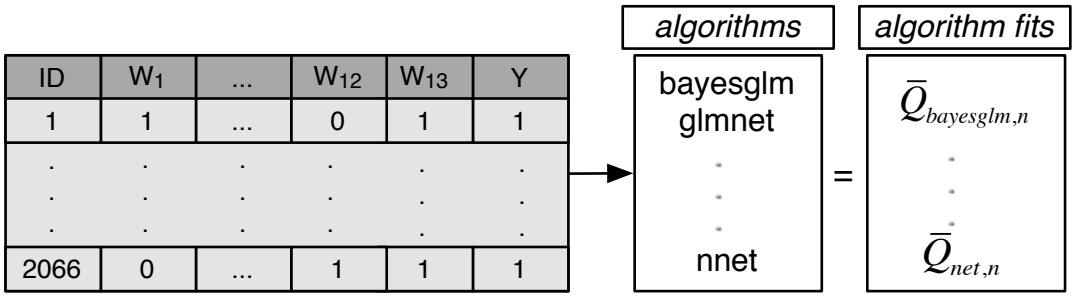
$$CV\ MSE_j = \frac{\sum_{i=1}^{n}(Y_i - D_{j,i})^2}{n}$$

**6.** Calculate the optimal weighted combination of $M$ algorithms from a family of weighted combinations indexed by the weight vector $\alpha$. This is done by performing a regression of $Y$ on the predicted values $D$ to estimate the vector $\alpha$. This calculation determines the combination that minimizes the CV risk over the family of weighted combinations.

$$P_n(Y=1 \mid D) = \text{expit}(\alpha_{bayesglm,n} D_{bayesglm} + \ldots + \alpha_{nnet,n} D_{nnet})$$

# SPPARCS Data: Super Learner

**7.** Fit each of the *M* algorithms on the complete data set. These fits combined with the estimated weights form the super learner function that can be used for prediction.



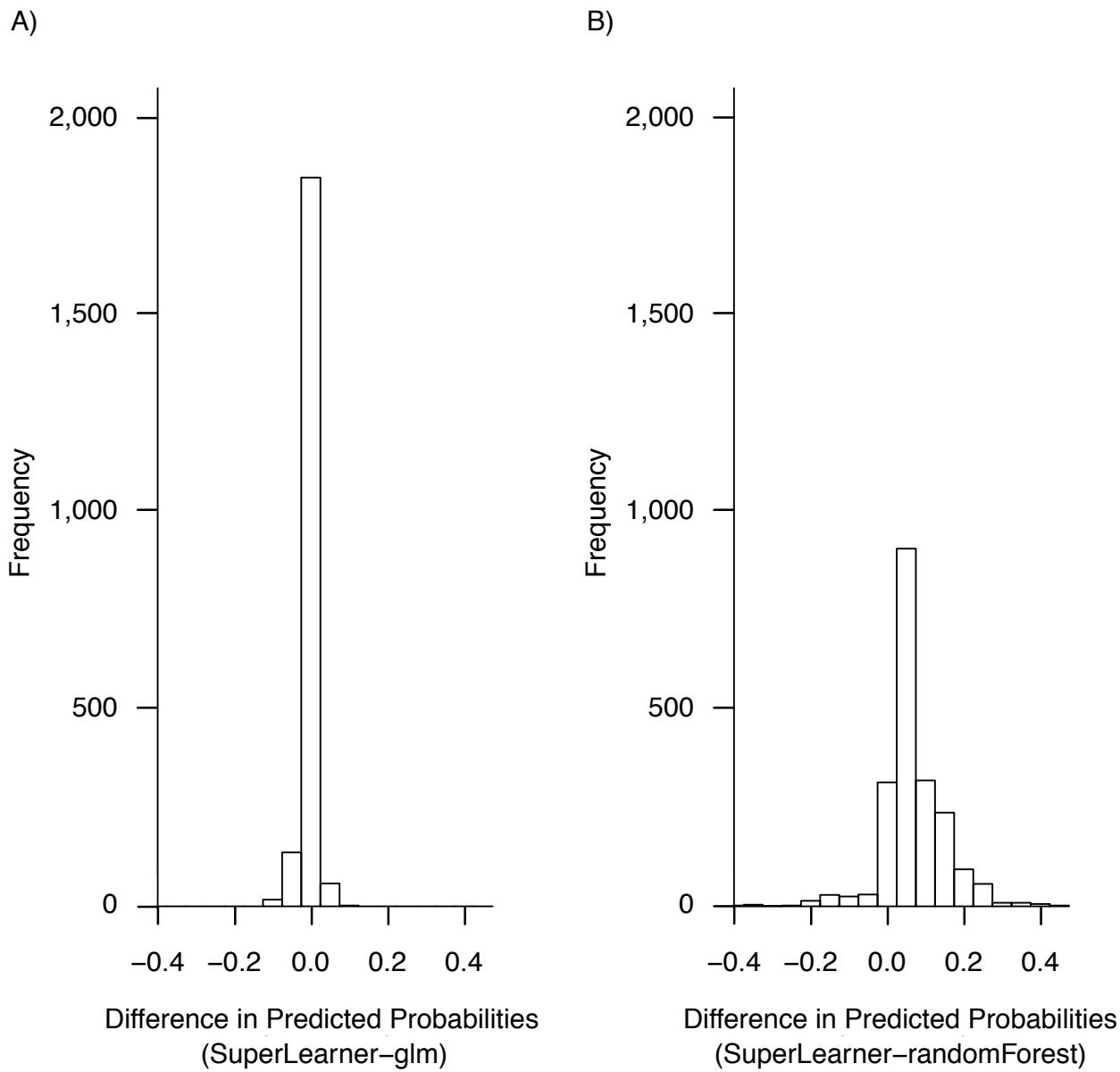

---

**8.** To obtain predicted values for the SPPARCS data, run the data through the super learner function.

$$\bar{Q}_{SL,n} = 0.461\bar{Q}_{bayesglm,n} + 0.496\bar{Q}_{gbm,n} + 0.044\bar{Q}_{mean,n}$$

| Algorithm | $RE$ | $R^2$ |
|---|---|---|
| `SuperLearner` | - | 0.201 |
| `glmnet` | 1.01 | 0.195 |
| `gbm` | 1.01 | 0.195 |
| `bayesglm` | 1.01 | 0.195 |
| `gam` | 1.01 | 0.194 |
| `glm` | 1.01 | 0.194 |
| `polymars` | 1.05 | 0.159 |
| `earth` | 1.06 | 0.157 |
| `ipredbagg` | 1.09 | 0.131 |
| `randomForest` | 1.12 | 0.105 |
| `rpart` | 1.15 | 0.078 |
| `mean` | 1.25 | 0.000 |
| `nnet` | 1.44 | -0.150 |

# Predicted Values



# Example Discussion

Super learner outperformed all algorithms in the collection of algorithms, although its performance was similar to several of the algorithms.

Even when the result is a negligible improvement relative to the best algorithms in the collection, the super learner provides a tool to run many algorithms and return a prediction function with the best (or equal) cross-validated MSE, avoiding the need to commit to a single algorithm.

# Oracle Properties

We now introduce the concept of the oracle selector, which is the best estimator given the $K$ algorithms in the library of algorithms. The oracle selector chooses the algorithm with the smallest risk under $P_0$, the true probability distribution of the random variable $O$. However, the oracle selector is unknown since it depends on both the observed data and $P_0$.

# Oracle Properties

Theory shows that that the cross-validation selector performs as well as the oracle selector, up to a second order term. The loss function must be bounded, and then we will perform asymptotically as well as the algorithm selected by the oracle selector. The number of algorithms in the library can grow with sample size.

# More examples

To study the super learner in real data examples, we collected a number of publicly available data sets.

- sample sizes ranged from 200 to 654 observations
- number of covariates ranged from 3 to 18
- all 13 data sets have a continuous outcome and no missing values

# Finite sample performance

**Table 3.3** Description of data sets, where $n$ is the sample size and $p$ is the number of covariates. All examples have a continuous outcome.

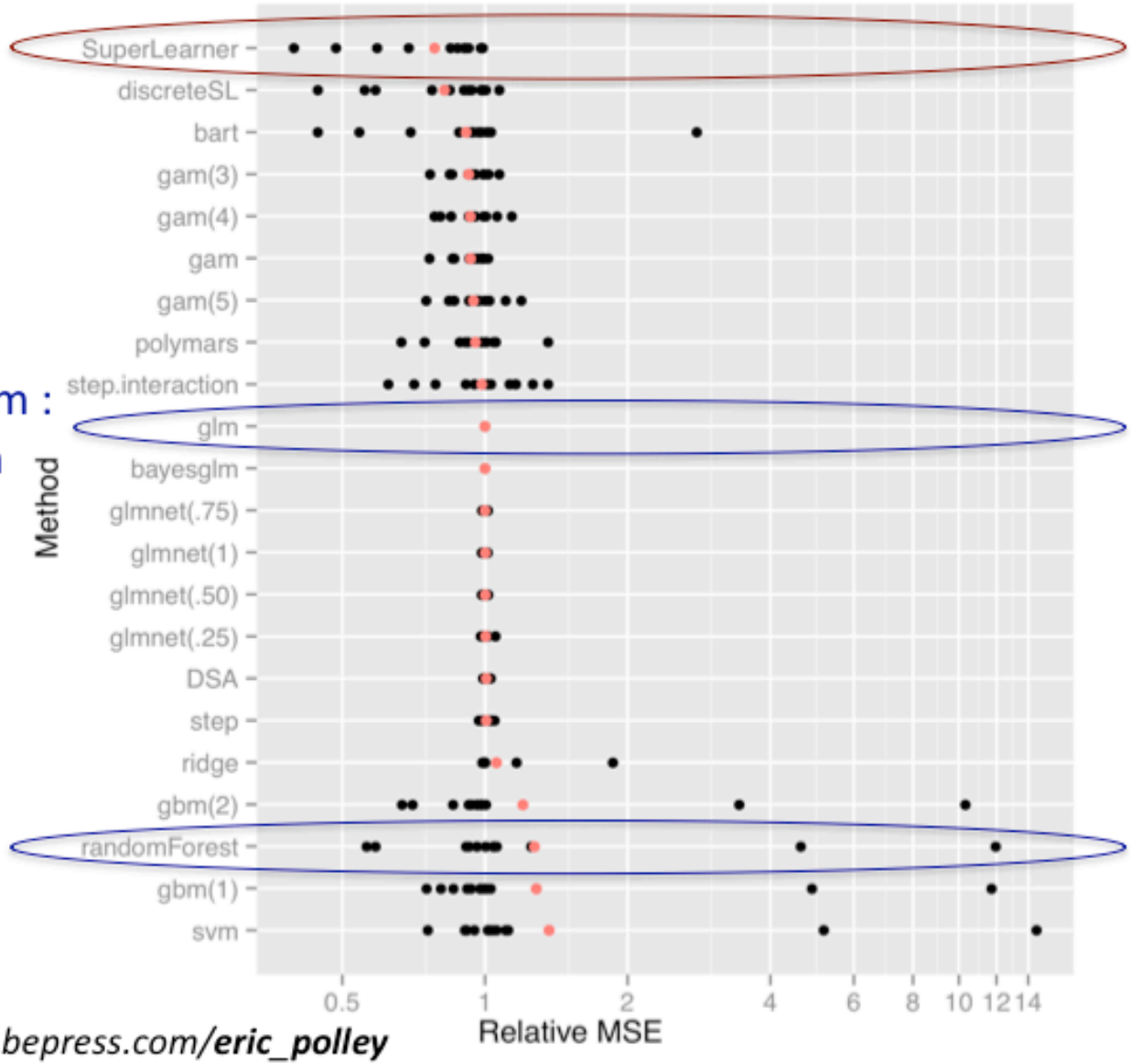
| Name | $n$ | $p$ | Source |
|---|---|---|---|
| ais | 202 | 10 | Cook and Weisberg (1994) |
| diamond | 308 | 17 | Chu (2001) |
| cps78 | 550 | 18 | Berndt (1991) |
| cps85 | 534 | 17 | Berndt (1991) |
| cpu | 209 | 6 | Kibler et al. (1989) |
| FEV | 654 | 4 | Rosner (1999) |
| Pima | 392 | 7 | Newman et al. (1998) |
| laheart | 200 | 10 | Afifi and Azen (1979) |
| mussels | 201 | 3 | Cook (1998) |
| enroll | 258 | 6 | Liu and Stengos (1999) |
| fat | 252 | 14 | Penrose et al. (1985) |
| diabetes | 366 | 15 | Harrell (2001) |
| house | 506 | 13 | Newman et al. (1998) |

# Finite sample performance

Super Learner- Best weighted combination of algorithms for a given prediction problem

Example algorithm : Linear Main Term Regression

Example algorithm: Random Forest




Technical Report: *works.bepress.com/**eric_polley***

# Finite sample performance

The super learner **outperforms the discrete super learner**, and **both outperform any individual algorithm.**

Among the individual algorithms, the Bayesian additive regression trees performs the best, but over-fits one of the data sets with a relative mean squared error of almost 3.0.

# Screening

In high dimensional data, it is often beneficial to screen the variables before running prediction algorithms.

Screening algorithms can be coupled with prediction algorithms to create new algorithms in the library.

For example:

- We may consider and algorithm using all features and an algorithm on the subset of only clinical variables. These two algorithms are considered unique algorithms.
- Another screening algorithm involves testing the pairwise correlations of each variable with the outcome and ranking the variables by the corresponding p-value.
- An additional screening algorithm involves running the glmnet algorithm and selecting the variables with non-zero coefficients.

# The free lunch

- There is no point in painstakingly trying to decide what estimators to enter in the collection; **instead add them all.**
- The theory supports this approach and finite sample simulations and data analyses only confirm that **it is very hard to over-fit the super learner by augmenting the collection**, but benefits are obtained.
- Indeed, for large data sets, we simply do not have enough algorithms available to build the desired collection that would fully utilize the power of the super learning.

# Additional References

- The super learner is a generalization of the stacking algorithm (Wolpert 1992, Breiman 1996) and has optimality properties that led to the name "super" learner.
- LeBlanc & Tibshirani (1996) discussed the relationship of stacking algorithms to model-mix algorithms (Stone 1974) and predictive sample-reuse methods (Geisser 1974).
- We refer to Chapter 3 in *Targeted Learning* for more comprehensive references.
- Rose, Fireman, van der Laan. Nested case-control risk score prediction. In: van der Laan, Rose. *Targeted Learning: Causal Inference for Observational and Experimental Data.* New York, NY: Springer, 2011:239-245.
- Rose. Mortality risk score prediction in an elderly population using machine learning. *Am J Epid*, in press.