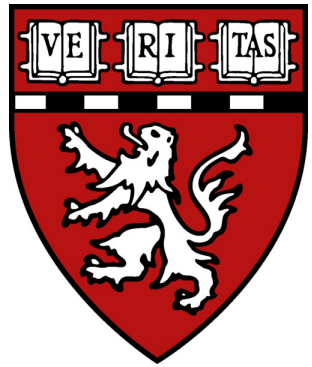
Machine Learning: Day 1

Sherri Rose

Associate Professor
Department of Health Care Policy
Harvard Medical School

drsherrirose.com
@sherrirose

February 27, 2017



Goals: Day 1

1. Understand shortcomings of standard parametric regression-based techniques for the estimation of prediction quantities.
2. Be introduced to the ideas behind machine learning approaches as tools for confronting the curse of dimensionality.
3. Become familiar with the properties and basic implementation of the super learner for prediction.

[Motivation]

Open access, freely available online

Essay

Why Most Published Research Findings Are False

John P. A. Ioannidis

The New York Times
nytimes.com

September 16, 2007

Do We Really Know What Makes Us Healthy?

By GARY TAUBES

Open access, freely available online

Essay

Why Most Published Research Findings Are False

John P. A. Ioannidis

The New York Times
nytimes.com

September 16, 2007

Do We Really Know What Makes Us Healthy?

By GARY TAUBES

---

4 high impact zones in statistical discovery with big data

FierceBigData

September 22, 2014

SHARE

Email

38 Tweet

23 Share

4 Like

2 +1

Clockwise from top left: Dunson, Rudin, McCormic and Rose

By: Sherri Rose, Harvard University; David Dunson, Duke University; Tyler McCormick, University of Washington; and Cynthia Rudin, MIT

Big data is transforming society with the help of statisticians, who possess in-depth experience and expertise in the art and science surrounding data. The American Statistical Association, or ASA, has recently released a white paper entitled "Discovery with Data: Leveraging Statistics with Computer Science to Transform Science and Society" (pdf) that highlights high-impact areas where statistical science is being applied to transformative big data research questions. Statistics is, by definition, the science of learning from data, and has had a key impact in several of the most prominent fields of discovery, including the biological sciences, health care, business analytics and recommendation systems, and the social sciences. There is a strong need to work in integrated teams comprised of domain experts, statisticians, and computer scientists in order to solve these complicated problems, which require tailored solutions using the influx of big data.

---

variations

Big data and the future

At the beginning of her career **Sherri Rose** discusses big data and stands amazed at its potential.

STATtr@k *A website for new statistics professionals navigating a data-centric*

Home | About Us | ASA Membership | Get Involved | Awards & Scholarships | Career

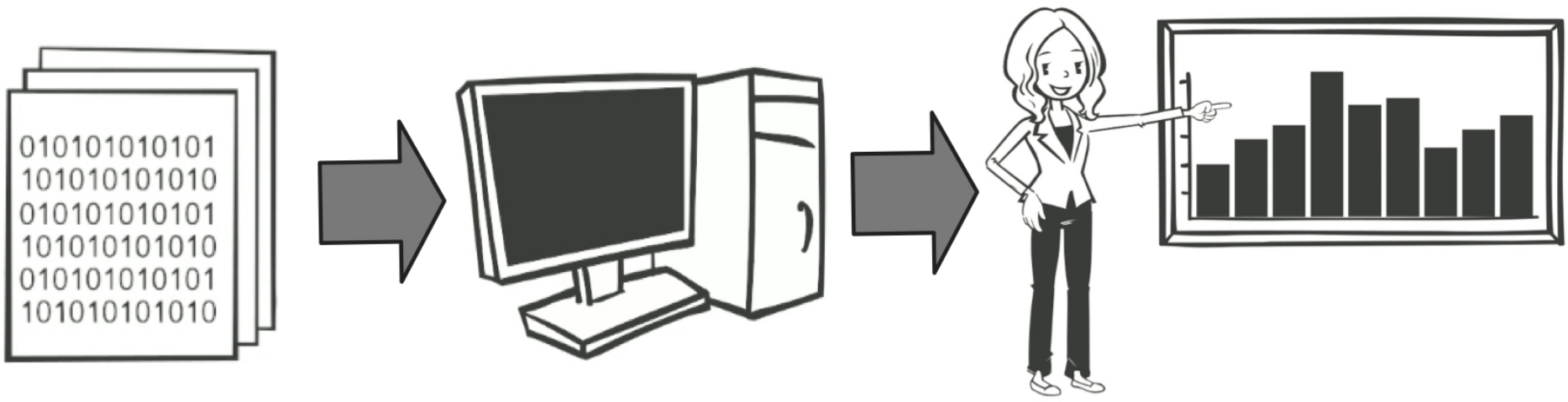
Statisticians' Place in Big Data

– FEBRUARY 1, 2013
POSTED IN: DEVELOPMENT TR@K

Sherri Rose is an NSF mathematical sciences postdoctoral research fellow in the department of biostatistics at the Johns Hopkins Bloomberg School of Public Health.

Big Data has become the new buzz phrase in the world of information collection and analysis. The experiments we conduct and the observational data we collect continue to grow in size, due to rapidly expanding technology.

Large data sets also have drawn the attention of young people, with undergraduate and graduate students choosing computer science, engineering, and statistics for their programs of study. Each of these disciplines brings something unique to the table when discussing the challenges of Big Data, and interdisciplinary collaborations are becoming increasingly common.


010101010101
101010101010
010101010101
101010101010
010101010101
101010101010

Electronic Health Databases

The increasing availability of electronic medical records offers a **new resource to public health researchers**.

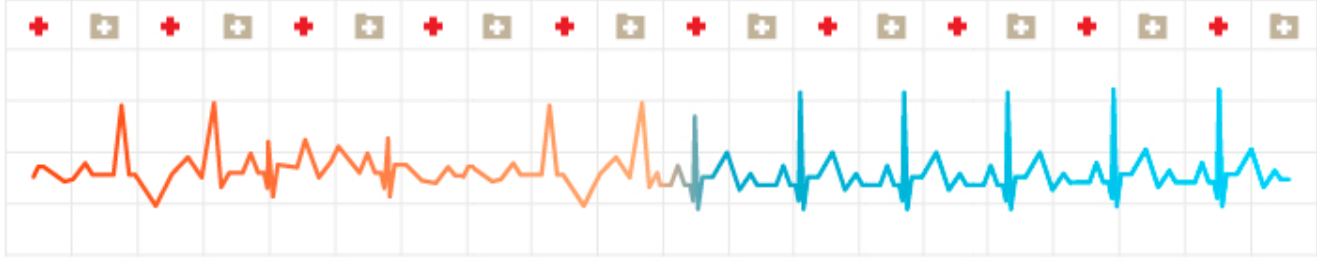
General usefulness of this type of data to answer targeted scientific research questions is an open question.

Need **novel statistical methods** that have desirable statistical properties while remaining computationally feasible.

Electronic Health Databases

- FDA's **Sentinel Initiative** aims to monitor drugs and medical devices for safety over time **already has access to 100 million people and their medical records**.

- The **$3 million Heritage Health Prize Competition** where the goal was to predict future hospitalizations using existing high-dimensional patient data.



Improve Healthcare,
Win $3,000,000.

Electronic Health Databases

- **Truven MarketScan** database. Contains information on enrollment and claims from private health plans and employers.

TRUVEN HEALTH ANALYTICS
More Than Data. Answers.
MARKETSCAN® RESEARCH

- **Health Insurance Marketplace** has enrolled over 10 million people.

High Dimensional 'Big Data' Parametric Regression

- Often dozens, hundreds, or even thousands of potential variables

1513	4.163950	3.854244	3.827496
1555	4.277033	3.373982	489.825226
1597	4.390150	3.795142	221.608444
1639	4.503117	3.640379	26.986557
1681	4.616217	3.336954	104.501778
1723	4.729317	3.561723	8.354190
1765	4.842267	3.576960	146.476227
1807	4.955350	3.858309	58.118893
1849	5.068450	3.514176	3.682388
1891	5.181567	3.794615	32.864357
1933	5.294517	3.311670	1.653655
1975	5.407600	3.931615	72.284065
2017	5.520700	4.319901	15.170299
2059	5.633650	3.938955	2.626603
2101	5.746750	3.924497	16.581503
2143	5.859883	3.771340	33.761124
2185	5.972850	3.797512	9.262811
2227	6.085967	3.795501	126.762199
2269	6.199067	3.759673	108.416565
2311	6.312167	3.373145	10.712665
2353	6.425117	3.464702	56.385990
2395	6.538183	3.640879	30.747551
2437	6.651333	3.702649	5.748046
2479	6.764283	3.941036	58.997993
2521	6.877350	3.393778	24.935211
2563	6.990450	3.213435	6.881421
2605	7.103400	3.635089	12.697396
2647	7.216517	3.749416	4.405899
2689	7.329650	3.450428	6.340690
2731	7.442750	3.287580	231.588028

High Dimensional 'Big Data' Parametric Regression

- Often dozens, hundreds, or even thousands of potential variables
- Impossible challenge to correctly specify the parametric regression

1513	4.163950	3.854244	3.827496
1555	4.277033	3.373982	489.825226
1597	4.390150	3.795142	221.608444
1639	4.503117	3.640379	26.986557
1681	4.616217	3.336954	104.501778
1723	4.729317	3.561723	8.354190
1765	4.842267	3.576960	146.476227
1807	4.955350	3.858309	58.118893
1849	5.068450	3.514176	3.682388
1891	5.181567	3.794615	32.864357
1933	5.294517	3.311670	1.653655
1975	5.407600	3.931615	72.284065
2017	5.520700	4.319901	15.170299
2059	5.633650	3.938955	2.626603
2101	5.746750	3.924497	16.581503
2143	5.859883	3.771340	33.761124
2185	5.972850	3.797512	9.262811
2227	6.085967	3.795501	126.762199
2269	6.199067	3.759673	108.416565
2311	6.312167	3.373145	10.712665
2353	6.425117	3.464702	56.385990
2395	6.538183	3.640879	30.747551
2437	6.651333	3.702649	5.748046
2479	6.764283	3.941036	58.997993
2521	6.877350	3.393778	24.935211
2563	6.990450	3.213435	6.881421
2605	7.103400	3.635089	12.697396
2647	7.216517	3.749416	4.405899
2689	7.329650	3.450428	6.340690
2731	7.442750	3.287580	231.588028

High Dimensional 'Big Data' Parametric Regression

- Often dozens, hundreds, or even thousands of potential variables
- Impossible challenge to correctly specify the parametric regression
- May have more unknown parameters than observations

1513	4.163950	3.854244	3.827496
1555	4.277033	3.373982	489.825226
1597	4.390150	3.795142	221.608444
1639	4.503117	3.640379	26.986557
1681	4.616217	3.336954	104.501778
1723	4.729317	3.561723	8.354190
1765	4.842267	3.576960	146.476227
1807	4.955350	3.858309	58.118893
1849	5.068450	3.514176	3.682388
1891	5.181567	3.794615	32.864357
1933	5.294517	3.311670	1.653655
1975	5.407600	3.931615	72.284065
2017	5.520700	4.319901	15.170299
2059	5.633650	3.938955	2.626603
2101	5.746750	3.924497	16.581503
2143	5.859883	3.771340	33.761124
2185	5.972850	3.797512	9.262811
2227	6.085967	3.795501	126.762199
2269	6.199067	3.759673	108.416565
2311	6.312167	3.373145	10.712665
2353	6.425117	3.464702	56.385990
2395	6.538183	3.640879	30.747551
2437	6.651333	3.702649	5.748046
2479	6.764283	3.941036	58.997993
2521	6.877350	3.393778	24.935211
2563	6.990450	3.213435	6.881421
2605	7.103400	3.635089	12.697396
2647	7.216517	3.749416	4.405899
2689	7.329650	3.450428	6.340690
2731	7.442750	3.287580	231.588028

High Dimensional 'Big Data' Parametric Regression

- Often dozens, hundreds, or even thousands of potential variables
- Impossible challenge to correctly specify the parametric regression
- May have more unknown parameters than observations
- True functional might be described by a complex function not easily approximated by main terms or interaction terms

1513	4.163950	3.854244	3.827496
1555	4.277033	3.373982	489.825226
1597	4.390150	3.795142	221.608444
1639	4.503117	3.640379	26.986557
1681	4.616217	3.336954	104.501778
1723	4.729317	3.561723	8.354190
1765	4.842267	3.576960	146.476227
1807	4.955350	3.858309	58.118893
1849	5.068450	3.514176	3.682388
1891	5.181567	3.794615	32.864357
1933	5.294517	3.311670	1.653655
1975	5.407600	3.931615	72.284065
2017	5.520700	4.319901	15.170299
2059	5.633650	3.938955	2.626603
2101	5.746750	3.924497	16.581503
2143	5.859883	3.771340	33.761124
2185	5.972850	3.797512	9.262811
2227	6.085967	3.795501	126.762199
2269	6.199067	3.759673	108.416565
2311	6.312167	3.373145	10.712665
2353	6.425117	3.464702	56.385990
2395	6.538183	3.640879	30.747551
2437	6.651333	3.702649	5.748046
2479	6.764283	3.941036	58.997993
2521	6.877350	3.393778	24.935211
2563	6.990450	3.213435	6.881421
2605	7.103400	3.635089	12.697396
2647	7.216517	3.749416	4.405899
2689	7.329650	3.450428	6.340690
2731	7.442750	3.287580	231.588028

Estimation is a Science

1. **Data**: realizations of random variables with a probability distribution.
2. **Statistical Model**: actual knowledge about the shape of the data-generating probability distribution.
3. **Statistical Target Parameter**: a feature/function of the data-generating probability distribution.
4. **Estimator**: an a priori-specified algorithm, benchmarked by a dissimilarity-measure (e.g., MSE) w.r.t. target parameter.

Data

Random variable O, observed n times, could be defined in a simple case as $O = (W, A, Y) \sim P_0$ if we are without common issues such as missingness and censoring.

- W: vector of covariates
- A: exposure or treatment
- Y: outcome

This data structure makes for effective examples, but data structures found in practice are frequently more complicated.

Model

General case: Observe n i.i.d. copies of random variable O with probability distribution P_0.

The data-generating distribution P_0 is also known to be an element of a statistical model $\mathcal{M}$: $P_0 \in \mathcal{M}$.

A **statistical model** $\mathcal{M}$ is the set of possible probability distributions for P_0; it is a collection of probability distributions.

If all we know is that we have n i.i.d. copies of O, this can be our statistical model, which we call a nonparametric statistical model

Effect Estimation vs. Prediction

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

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.

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.

[Prediction with Super Learning]

Prediction

Standard practice involves assuming a parametric statistical model & using maximum likelihood to estimate the parameters in that statistical model.

Prediction: The Goal

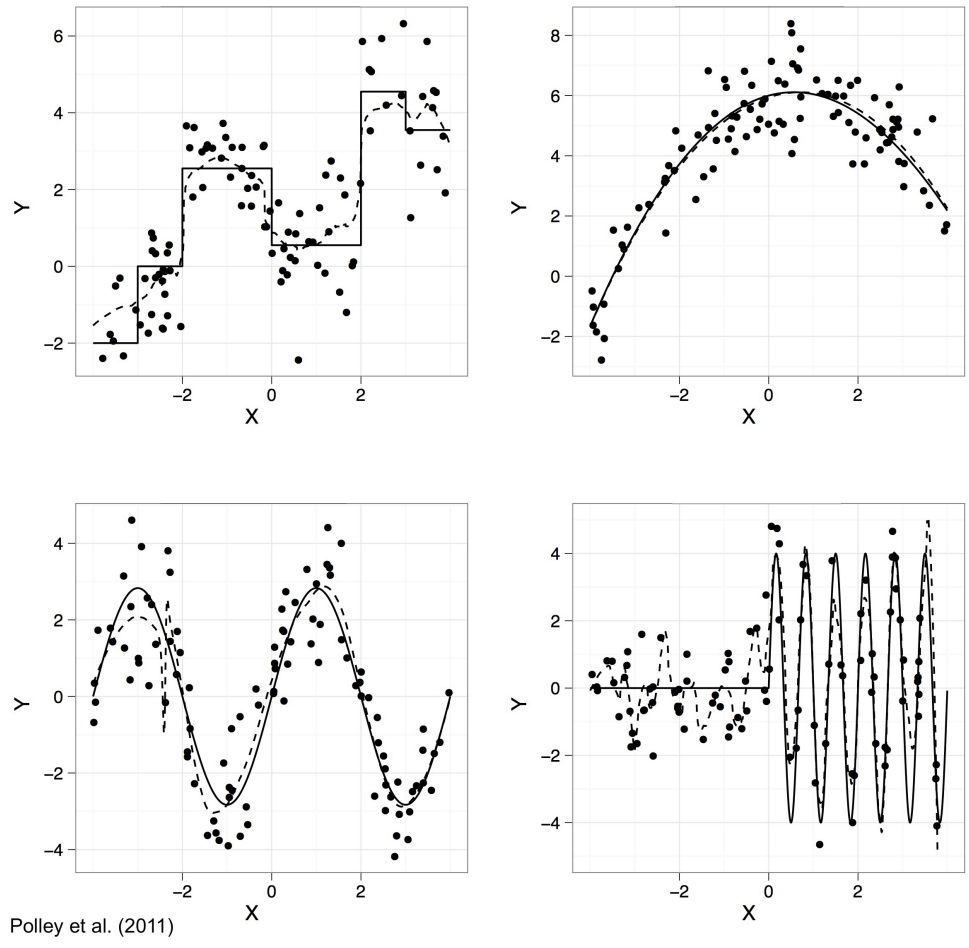
Flexible algorithm to estimate the regression function $E_0(Y \mid W)$.

Y outcome
W covariates

Prediction: Big Picture

Machine learning aims to

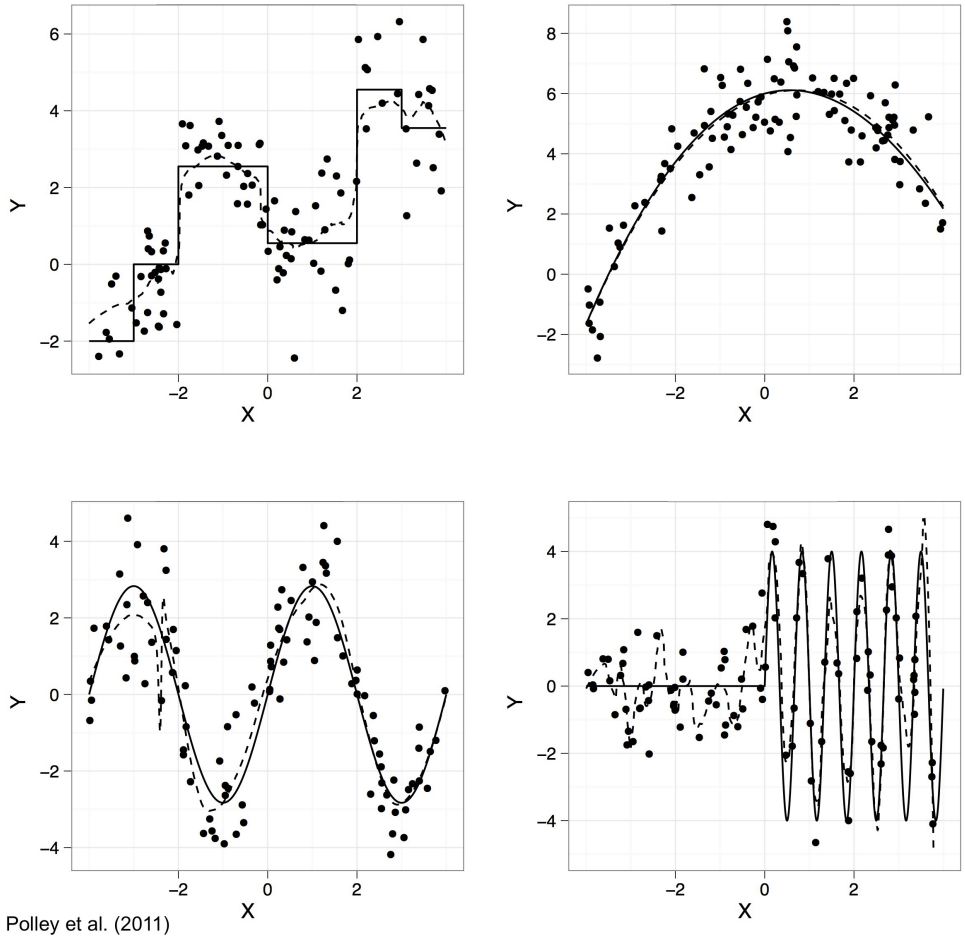
- "smooth" over the data
- make fewer assumptions



Prediction: Big Picture

Purely nonparametric model with high dimensional data?

- $p > n$!
- data sparsity



Polley et al. (2011)

Nonparametric Prediction Example: Local Averaging

- Local averaging of the outcome Y within covariate "neighborhoods."
- Neighborhoods are bins for observations that are close in value.
- The number of neighborhoods will determine the smoothness of our regression function.
- How do you choose the size of these neighborhoods?

Nonparametric Prediction Example: Local Averaging

- Local averaging of the outcome Y within covariate "neighborhoods."
- Neighborhoods are bins for observations that are close in value.
- The number of neighborhoods will determine the smoothness of our regression function.
- How do you choose the size of these neighborhoods?

This becomes a **bias-variance** trade-off question.

- Many small neighborhoods: high variance since some neighborhoods will be empty or contain few observations.
- Few large neighborhoods: biased estimates if neighborhoods fail to capture the complexity of data.

Prediction: A Problem

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

How will you know?

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

Prediction: Options?

- Recent studies for prediction have employed newer **algorithms.**
  (any mapping from data to a predictor)

Prediction: Options?

- Recent studies for prediction have employed newer **algorithms.**
- Researchers are then left with questions, e.g.,
  - *"When should I use random forest instead of standard regression techniques?"*

Prediction: Options?

- Recent studies for prediction have employed newer **algorithms.**
- Researchers are then left with questions, e.g.,
  - *"When should I use random forest instead of standard regression techniques?"*

ELSEVIER

Journal of Clinical Epidemiology 63 (2010) 1145–1155

Journal of Clinical Epidemiology

Logistic regression had superior performance compared with regression trees for predicting in-hospital mortality in patients hospitalized with heart failure

Peter C. Austin[a,c,*], Jack V. Tu[a,b,c,d,e], Douglas S. Lee[a,e,f]

Prediction: Options?

- Recent studies for prediction have employed newer **algorithms.**
- Researchers are then left with questions, e.g.,
  - *"When should I use random forest instead of standard regression techniques?"*

ELSEVIER

Journal of Clinical Epidemiology 63 (2010) 1145–1155

Journal of Clinical Epidemiology

Logistic regression had superior p
trees for predicting in-hospital n
hea

Peter C. Austin[a,c,*], Jack

European Journal of Neurology 2010, **17:** 945–950 doi:10.1111/j.1468-1331.2010.02955.x

Random forest can predict 30-day mortality of spontaneous intracerebral hemorrhage with remarkable discrimination

S. -Y. Peng[a,b,c], Y. -C. Chuang[b], T. -W. Kang[b] and K. -H. Tseng[d]

[a]*Institute of Biomedical Informatics, National Yang-Ming University, Taipei;* [b]*Department of Anesthesiology, Taichung Veterans General Hospital, Taichung;* [c]*School of Medicine, Chung Shan Medical University, Taichung; and* [d]*Department of Nephrology, Taoyuan Veterans Hospital, Taoyuan, Taiwan*

Prediction: Key Concepts

Loss-Based Estimation

Use **loss functions** to define best estimator of $E_0(Y \mid W)$ & evaluate it.

Cross Validation

Available data is partitioned to **train** and **validate** our estimators.

Flexible Estimation

Allow **data** to drive your **estimates**, but in an honest (cross validated) way.

These are detailed topics; we'll cover core concepts.

Loss-Based Estimation

Wish to estimate: $\bar{Q}_0 = E_0(Y \mid W)$.

In order to choose a "best" algorithm to estimate this regression function, must have a way to define what "best" means.

Do this in terms of a loss function.

Loss-Based Estimation

Data structure is $O = (W, Y) \sim P_0$, with empirical distribution P_n which places probability $1/n$ on each observed O_i, $i = 1, \ldots, n$.

Loss function assigns a measure of performance to a candidate function $\bar{Q} = E(Y \mid W)$ when applied to an observation O.

Formalizing the Parameter of Interest

We define our parameter of interest, $\bar{Q}_0 = E_0(Y \mid W)$, as the minimizer of the expected squared error loss:

$$\bar{Q}_0 = \arg\min_{\bar{Q}} E_0 L(O, \bar{Q}),$$

where $L(O, \bar{Q}) = (Y - \bar{Q}(W))^2$.

$E_0 L(O, \bar{Q})$, which we want to be small, evaluates the candidate $\bar{Q}$, and it is minimized at the optimal choice of Q_0. We refer to expected loss as the risk

Y: Outcome, W: Covariates

Loss-Based Estimation

We want estimator of the regression function $\bar{Q}_0$ that minimizes the expectation of the squared error loss function.

This makes sense intuitively; we want an estimator that has small bias and variance.

Ensembling: Cross-Validation

- Ensembling methods allow implementation of multiple algorithms.
- Do not need to decide beforehand which single technique to use; can use several by incorporating cross validation.

Image credit: Rose (2010, 2016)

Ensembling: Cross-Validation

- Ensembling methods allow implementation of multiple algorithms.
- Do not need to decide beforehand which single technique to use; can use several by incorporating **cross-validation**.

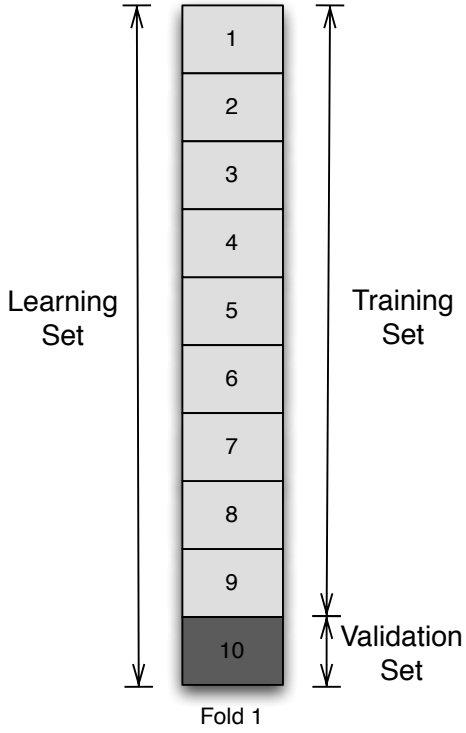


Image credit: Rose (2010, 2016)

Ensembling: Cross-Validation

- In V-fold cross-validation, our observed data $O_1, \ldots, O_n$ is referred to as the learning set and partition into V sets of size $\approx \frac{n}{V}$
- For any given fold, $V - 1$ sets comprise training set and remaining 1 set is validation set.

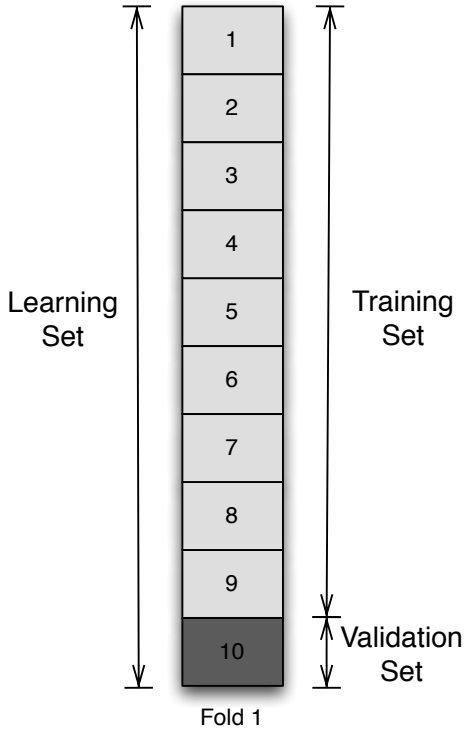


Image credit: Rose (2010, 2016)

Ensembling: Cross-Validation

- In V-fold cross-validation, our observed data $O_1, \ldots, O_n$ is referred to as the learning set and partition into V sets of size $\approx \frac{n}{V}$
- For any given fold, $V - 1$ sets comprise training set and remaining 1 set is validation set.

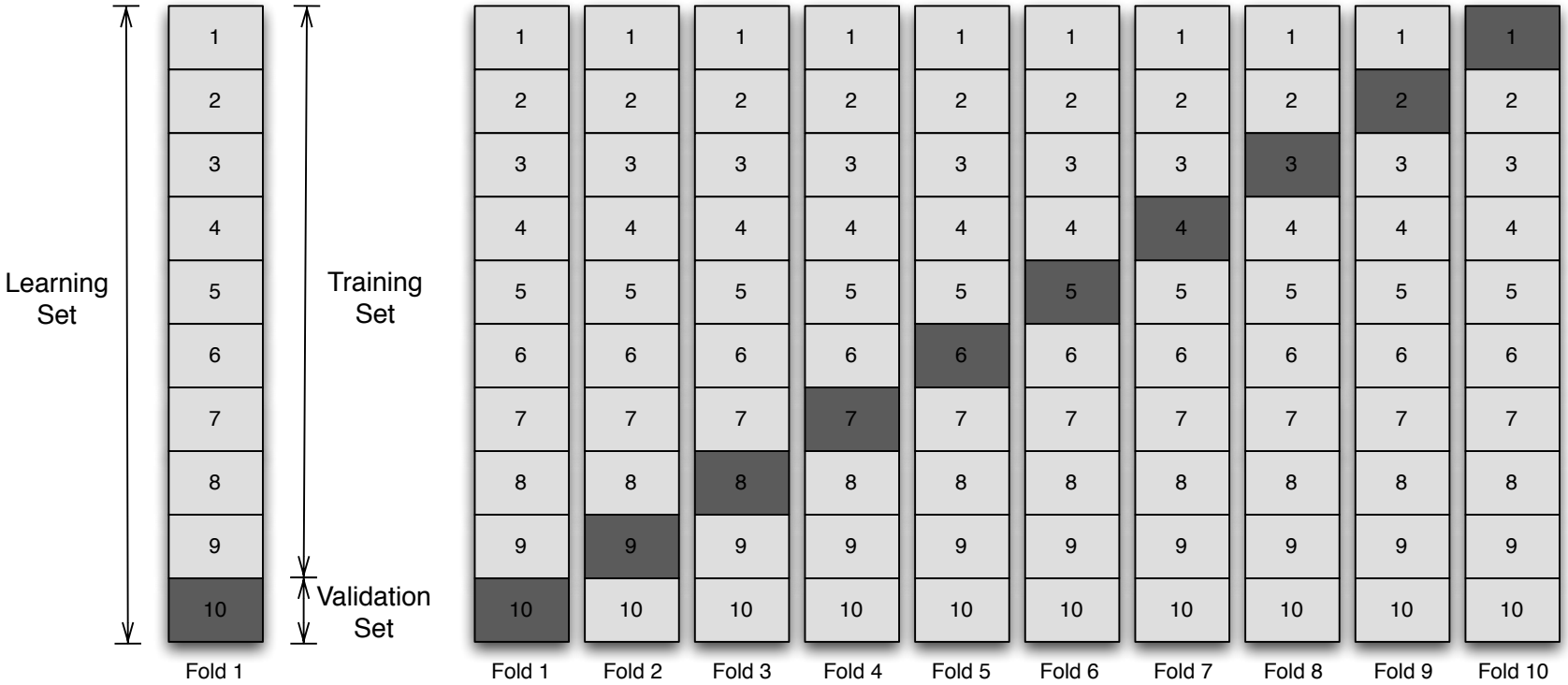


Image credit: Rose (2010, 2016)

Super Learner: Ensembling

Build a collection 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.

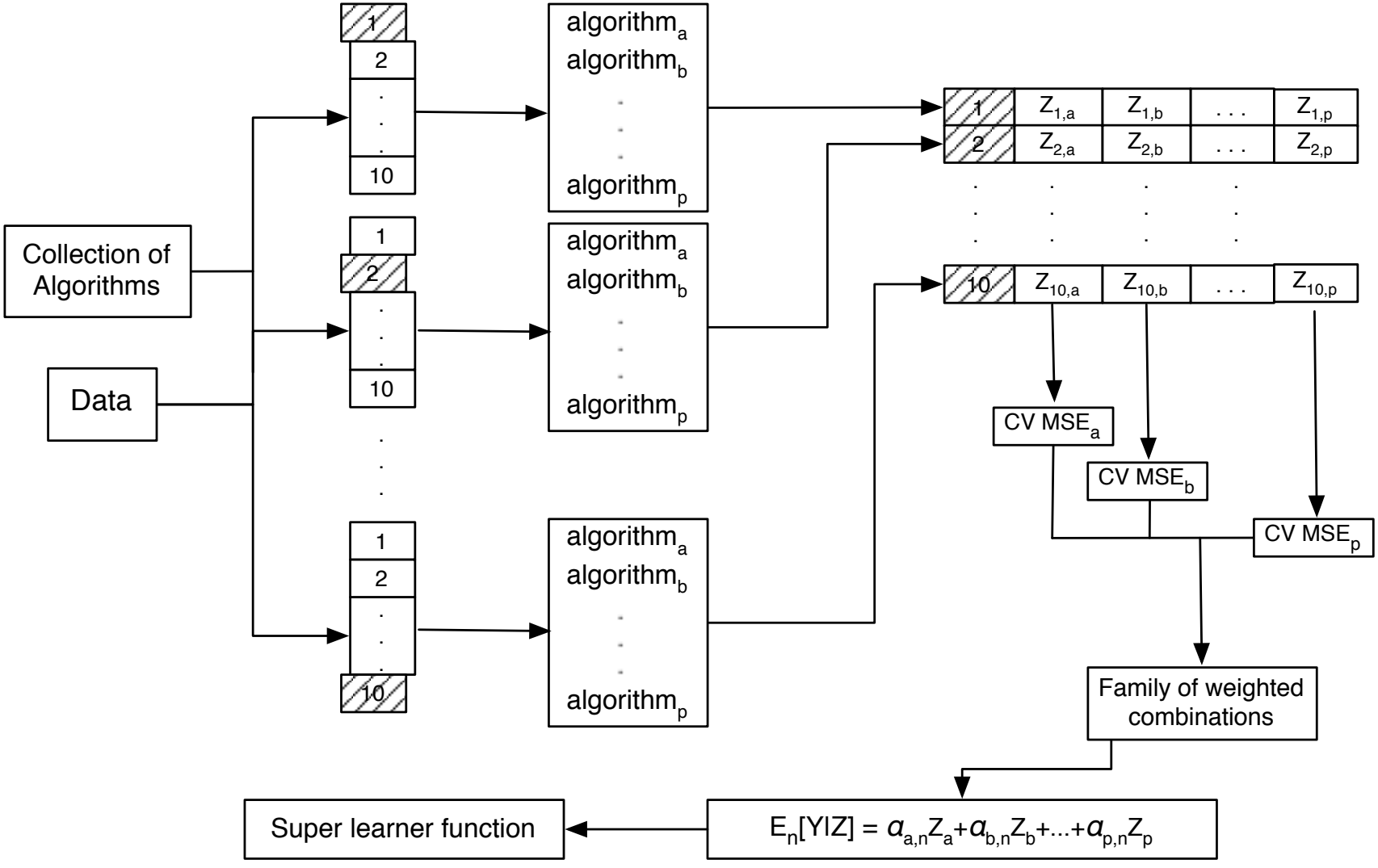


Image credit: Polley et al. (2011)

Super Learner: Optimal Weight Vector

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).

Super Learner: Optimal Weight Vector

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.*

Super Learner: Optimal Weight Vector

Consider that the discrete super learner has already been completed.

- Determine combination of algorithms that minimizes cross-validated risk.
- Propose family of weighted combinations of the algorithms, index by the weight vector α. The family of weighted combinations:
  - includes only those α-vectors that have a sum equal to one
  - each weight is positive or zero

Super Learner: Optimal Weight Vector

Consider that the discrete super learner has already been completed.

- Determine combination of algorithms that minimizes cross-validated risk.
- Propose family of weighted combinations of the algorithms, index by the weight vector α. The family of weighted combinations:
  - includes only those α-vectors that have a sum equal to one
  - each weight is positive or zero

Selecting the weights that minimize the cross-validated risk is a minimization problem, formulated as a regression of the outcomes Y on the predicted values of the algorithms (Z).

Super Learner: Optimal Weight Vector

Weight vector

$$E_n(Y \mid Z) = \alpha_{a,n} Z_a + \alpha_{b,n} Z_b + \ldots + \alpha_{p,n} Z_p$$

The (cross-validated) probabilities of the outcome (Z) for each algorithm are used as inputs in a working statistical model to predict the outcome Y.

Super Learner: Optimal Weight Vector

Weight vector

$$E_n(Y \mid Z) = \alpha_{a,n} Z_a + \alpha_{b,n} Z_b + \ldots + \alpha_{p,n} Z_p$$

We have a working model with multiple coefficients $\alpha = \{\alpha_a, \alpha_b, \ldots, \alpha_p\}$ that need to be estimated, one for each of the algorithms.

Super Learner: Optimal Weight Vector

Weight vector

$$E_n(Y \mid Z) = \alpha_{a,n} Z_a + \alpha_{b,n} Z_b + \ldots + \alpha_{p,n} Z_p$$

The weighted combination with the smallest cross-validated risk is the "best" estimator according to our criteria: minimizing the estimated expected squared error loss function.

Super Learner: Ensembling

Due to its theoretical properties, super learner:

performs asymptotically as well as the best choice among the family of weighted combinations of estimators.

Thus, **by adding more competitors, we only improve the performance of the super learner.**

The asymptotic equivalence remains true if the number of algorithms in the library grows very quickly with sample size.

Super Learner: Oracle Inequality

$B_n \in \{0,1\}^n$ splits the sample into a training sample $\{i : B_n(i) = 0\}$ and validation sample $\{i : B_n(i) = 1\}$. P^0_{n,B_n} and P^1_{n,B_n} denote the empirical distribution of the training and validation sample, respectively. Given candidate estimators $P_n \to \hat{Q}_k(P_n)$, the loss-function-based cross-validation selector is:

$$k_n = \hat{K}(P_n) = \arg\min_k E_{B_n} P^1_{n,B_n} L(\hat{Q}_k(P^0_{n,B_n})).$$

The resulting estimator is given by $\hat{Q}(P_n) = \hat{Q}_{\hat{K}(P_n)}(P_n)$ and satisfies the following oracle inequality: for any $\delta > 0$

$$E_{B_n}\{P_0 L(\hat{Q}_{k_n}(P^0_{n,B_n}) - L(Q_0)\} \leq (1+2\delta) E_{B_n} \min_k P_0\{L(\hat{Q}_k(P^0_{n,B_n})) - L(Q_0)\} + 2C(\delta)\frac{1+\log K(n)}{np}.$$

van der Laan & Dudoit (2003)

Screening: Will Be Useful for Parsimony

- Often beneficial to screen variables before running algorithms.
- Can be coupled with prediction algorithms to create new algorithms in the library.

Screening: Will Be Useful for Parsimony

- Often beneficial to screen variables before running algorithms.
- Can be coupled with prediction algorithms to create new algorithms in the library.
- Clinical subsets

Screening: Will Be Useful for Parsimony

- Often beneficial to screen variables before running algorithms.
- Can be coupled with prediction algorithms to create new algorithms in the library.
- Clinical subsets
- Test each variable with the outcome, rank by p-value

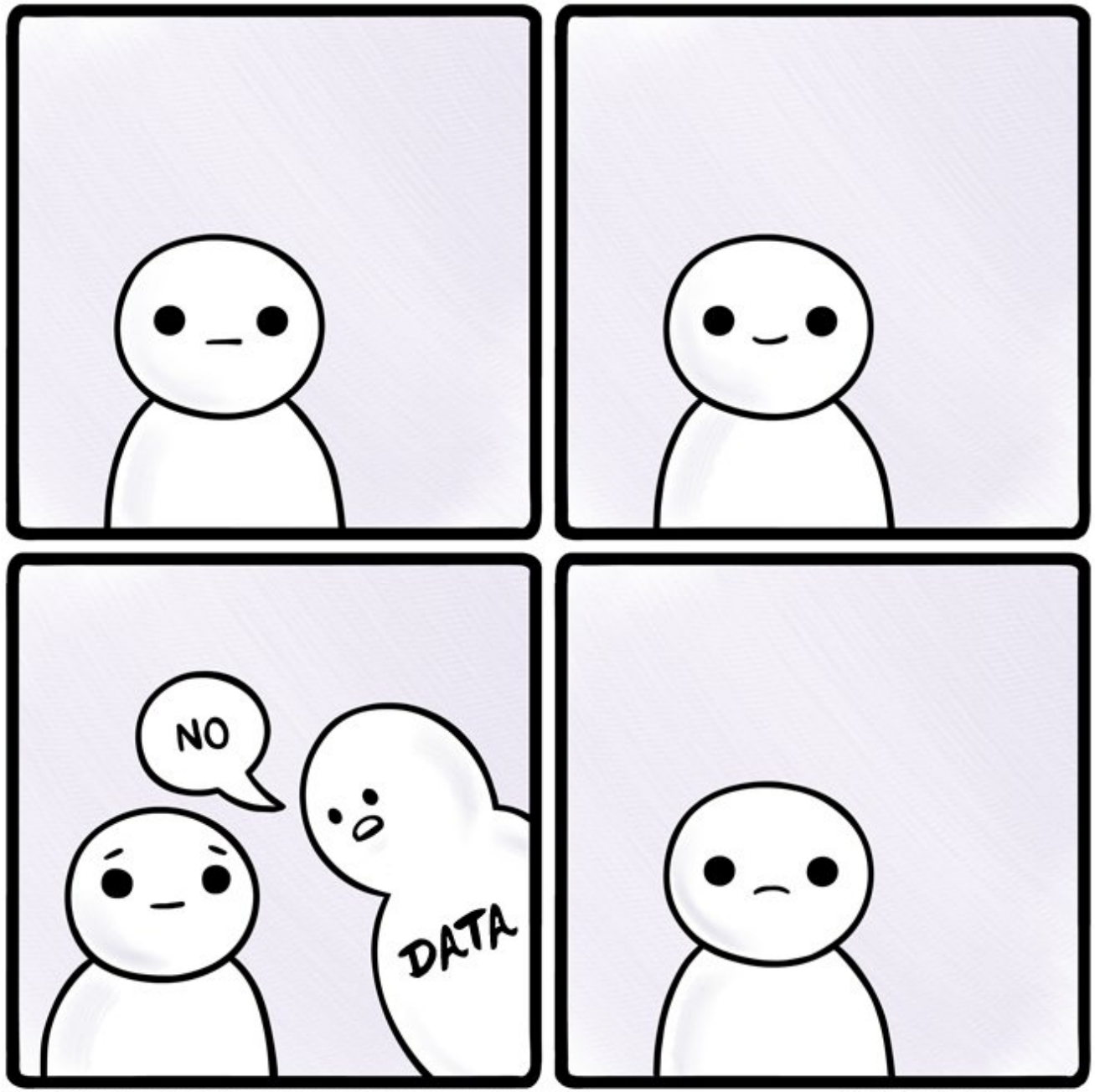
Screening: Will Be Useful for Parsimony

- Often beneficial to screen variables before running algorithms.
- Can be coupled with prediction algorithms to create new algorithms in the library.

- Clinical subsets
- Test each variable with the outcome, rank by p-value
- Lasso

The Free Lunch

- No point in painstakingly deciding which estimators; **add them all.**
- Theory supports this approach and finite sample simulations and data analyses only confirm that **it is very hard to overfit the super learner by augmenting the collection**, but benefits are obtained.



THIS COMIC MADE POSSIBLE THANKS TO ADAM LINGELBACH

MRLOVENSTEIN.COM

Mortality Risk Score Prediction in Elderly Populations

Previous studies 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).

Super Learner: Kaiser Permanente Database

Kaiser Permanente is based in Northern California and provides medical services to approximately 350,000 persons over the age of 65 each year.

- **Gender** & **age** obtained from administrative databases
- **184 disease and diagnoses variables (medical flags)** obtained from clinical and claims databases

Super Learner: Kaiser Permanente Database

Nested case-control sample (n=27,012).

- **Outcome**: death.
- **Covariates**: 184 medical flags, gender & age.

Ensembling method outperformed all other algorithms.

Generally weak signal with $R^2 = 0.11$.

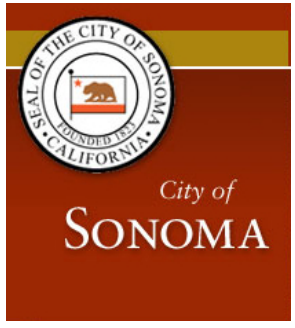
Observed data structure on a subject can be represented as $O = (Y, \Delta, \Delta X)$, where $X = (W, Y)$ is the full data structure, and Δ denotes the indicator of inclusion in the second-stage sample.

How will this electronic database perform in comparison to a cohort study?

van der Laan & Rose (2011)

Super Learner: Sonoma Cohort Study

- 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.



Super Learner: Sonoma Cohort Study

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.

Super Learner: Sonoma Cohort Study

Table: Characteristics ($n = 2,066$)

Variable	No.	%
Death (Y)	269	13
Female (W_1)	1,225	59
Age, years		
54 to 60 (W_2)	323	16
61 to 70 (W_3)	749	36
71 to 80	1,339	65
81 to 90 (W_4)	245	12
$>$ 90 (W_5)	22	11

Super Learner: Sonoma Cohort Study

Table: Characteristics ($n = 2,066$)

Variable	No.	%
Self-rated health, baseline		
excellent (W_6)	657	32
good	1,037	50
fair (W_7)	309	15
poor (W_8)	63	3
Met minimum physical activity level (W_9)	1,460	71
Current smoker (W_{10})	172	8
Former smoker (W_{11})	1,020	49
Cardiac event prior to baseline (W_{12})	356	17
Chronic health condition at baseline (W_{13})	918	44

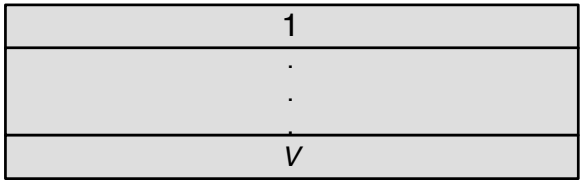
Super Learner: Sonoma Cohort Study

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

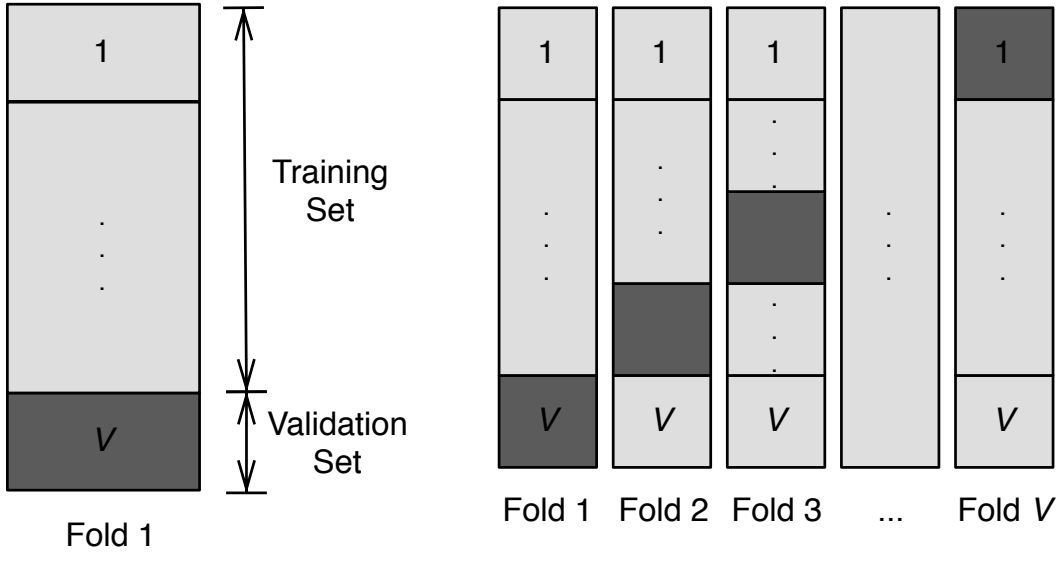
ID	W_1	...	W_{12}	W_{13}	Y
1	1	...	0	1	1
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
2066	0	...	1	1	1

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.



Super Learner: Sonoma Cohort Study

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.

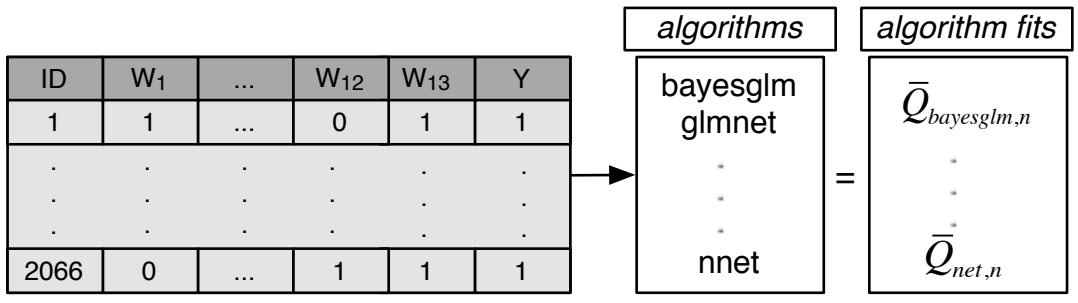
$$CV\ MSE_j = \frac{\sum_{i=1}^{n}(Y_i - D_{j,i})^2}{n}$$

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

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

Super Learner: Sonoma Cohort Study

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.

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

Super Learner: Sonoma Cohort Study

Cohort study of $n = 2,066$ residents of Sonoma, CA aged 54 and over.

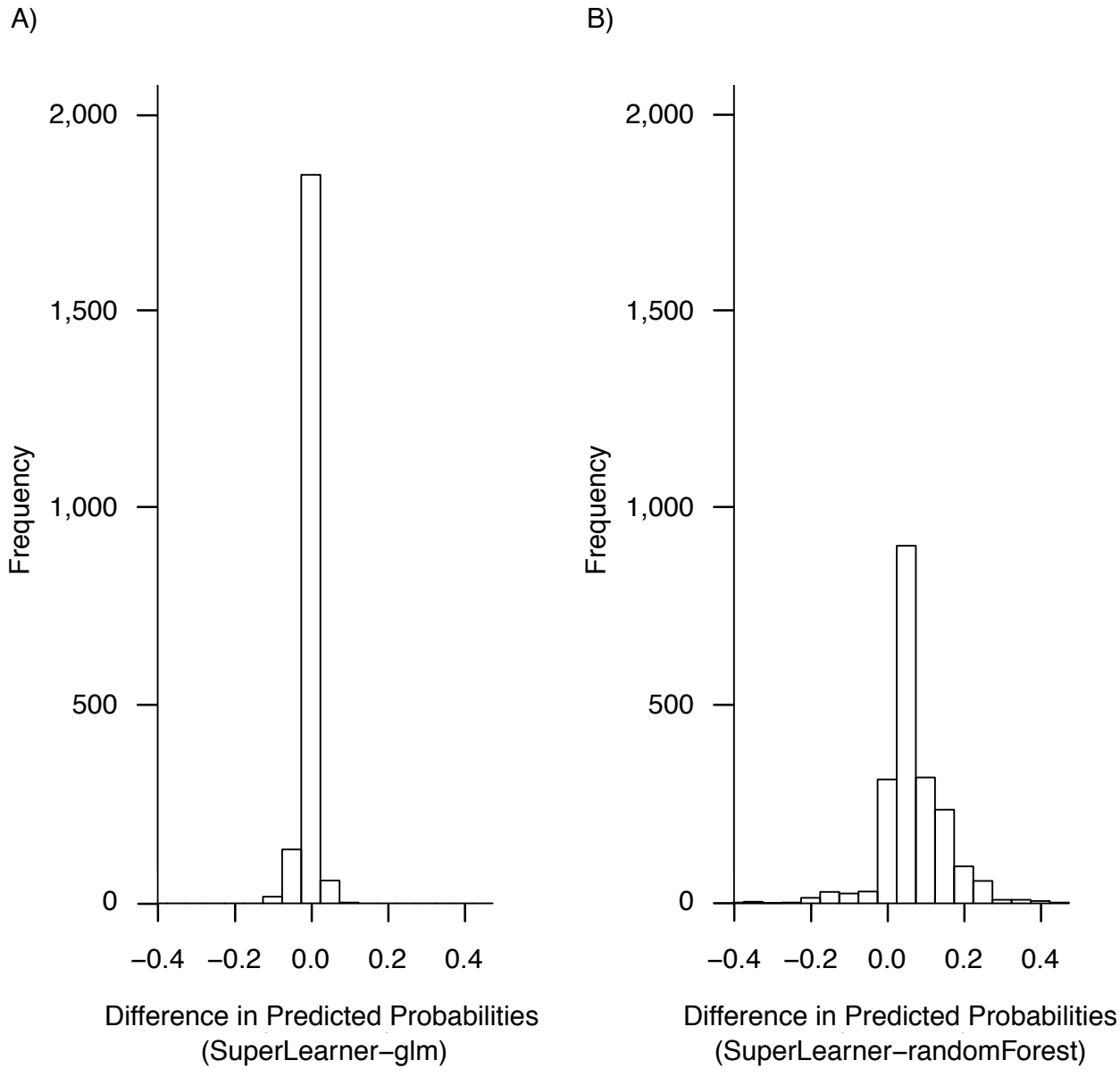
- Outcome: death.
- Covariates: gender, age, **self-rated health**, **leisure-time physical activity**, smoking status, cardiac event history, and chronic health condition status.
- $R^2 = 0.201$

Two-fold improvement with less than 10% of the subjects & less than 10% the number of covariates.

What possible conclusions can we draw?

Rose (2013)

Super Learner: Sonoma Cohort Study



Super Learner: Sonoma Cohort Study

- Previous literature indicates that perception of health in elderly adults may be as important as less subjective measures when assessing later outcomes (Idler & Benyamini 1997, Blazer 2008).
- Likewise, benefits of physical activity in older populations have also been shown (Denaei et al. 2009).

Super Learner: Public Datasets

Studied the super learner in 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

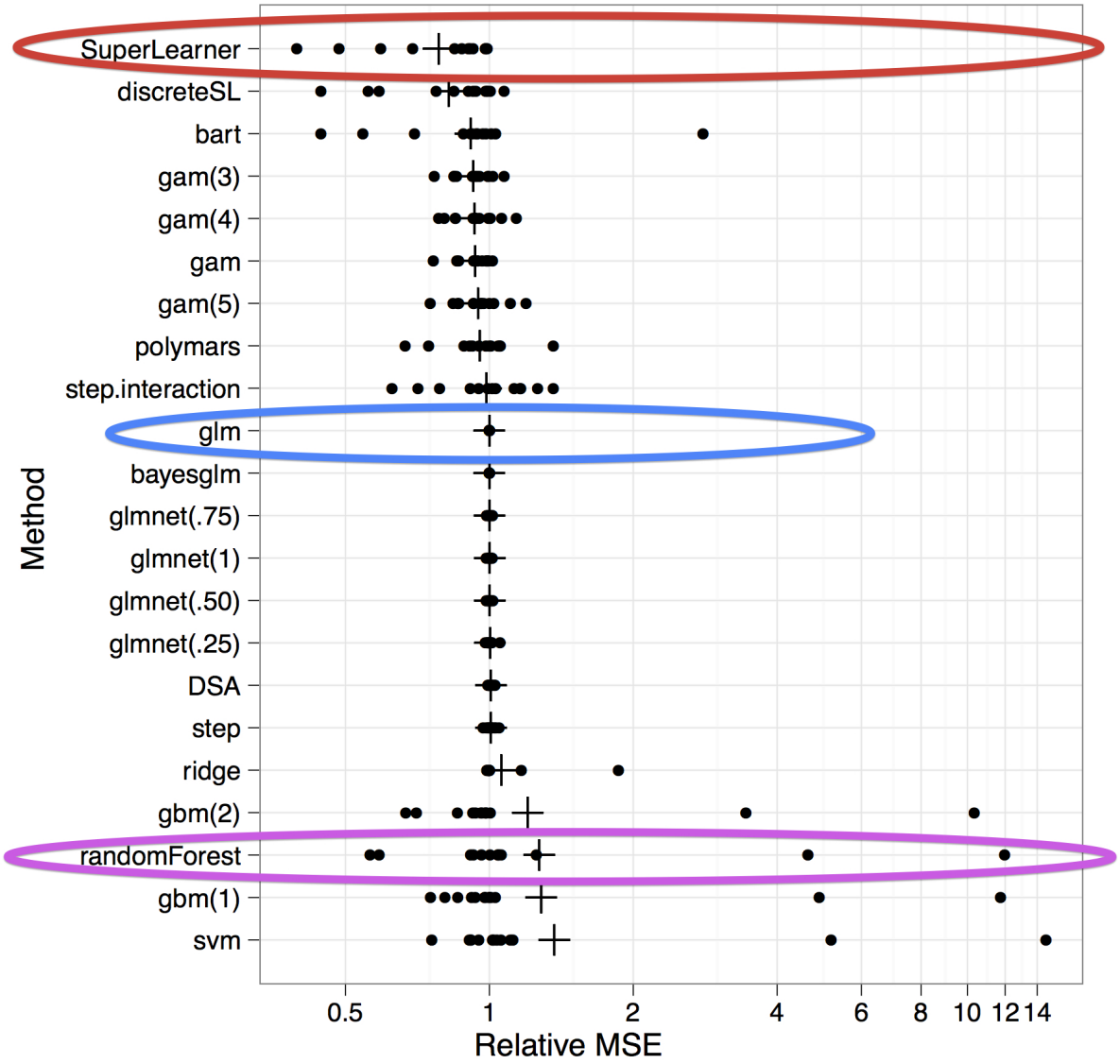
Polley et al. (2011)

Super Learner: Public Datasets

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)

Polley et al. (2011)

Super Learner: Public Datasets



Polley et al. (2011)

Super Learner: Mortality Risk Scores in ICUs

Risk scores for mortality in intensive care units is a difficult problem, and previous scoring systems did not perform well in validation studies.

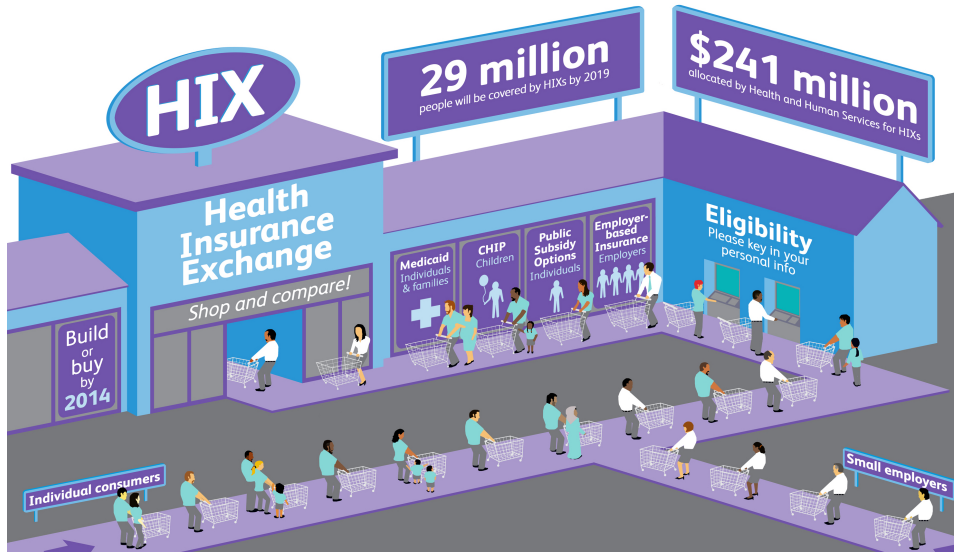
- Super learner had extraordinary performance with AUC of 94%
- Web interface

Pirracchio et al. (2015)

Super Learner: Plan Payment Implications

Over 50 million people in the United States currently enrolled in an insurance program that uses risk adjustment.

- Redistributes funds based on health
- Encourages competition based on efficiency/quality



xerox.com

Results

- Machine learning finds novel insights
- Potential to impact policy, including diagnostic upcoding and fraud

Health Insurance Marketplace

Rose (2016)

Super Learner: Predicting Unprofitability

- Take on role as hypothetical profit-maximizing insurer
- Health plan design on pre-existing conditions is now highly regulated in Health Insurance Marketplaces
- What about prescription drug offerings?

New super learner algorithm shows that this distortion is possible

Rose, Bergquist, Layton (2017)

Ensembling Literature

- 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 other algorithms.
- Additional methods for ensemble learning have also been developed (e.g., Tsybakov 2003; Juditsky et al. 2005; Bunea et al. 2006, 2007; Dalayan & Tsybakov 2007, 2008).
- Refer to a review of ensemble methods (Dietterich 2000) for further background.
- van der Laan et al. (2007) original super learner paper.
- For more references, see Chapter 3 of *Targeted Learning*.

[Super Learner Example Code]

Super Learner R Packages

- `SuperLearner` (Polley): Main super learner package
- `h2oEnsemble` (LeDell): Java-based, designed for big data, uses H2O `R` interface to run super learning
- `SAS` macro (Brooks): SAS implementation available on Github

More: `targetedlearningbook.com/software`

Super Learner Sample Code

```
install.packages("SuperLearner")
library(SuperLearner)
```

Super Learner Sample Code

```
##Generate simulated data##

set.seed(27)
n<-500
data <- data.frame(W1=runif(n, min = .5, max = 1),
W2=runif(n, min = 0, max = 1),
W3=runif(n, min = .25, max = .75),
W4=runif(n, min = 0, max = 1))
data <- transform(data,
W5=rbinom(n, 1, 1/(1+exp(1.5*W2-W3))))
data <- transform(data,
Y=rbinom(n, 1,1/(1+exp(-(-.2*W5-2*W1+4*W5*W1-1.5*W2+sin(W4))))))
```

Super Learner Sample Code

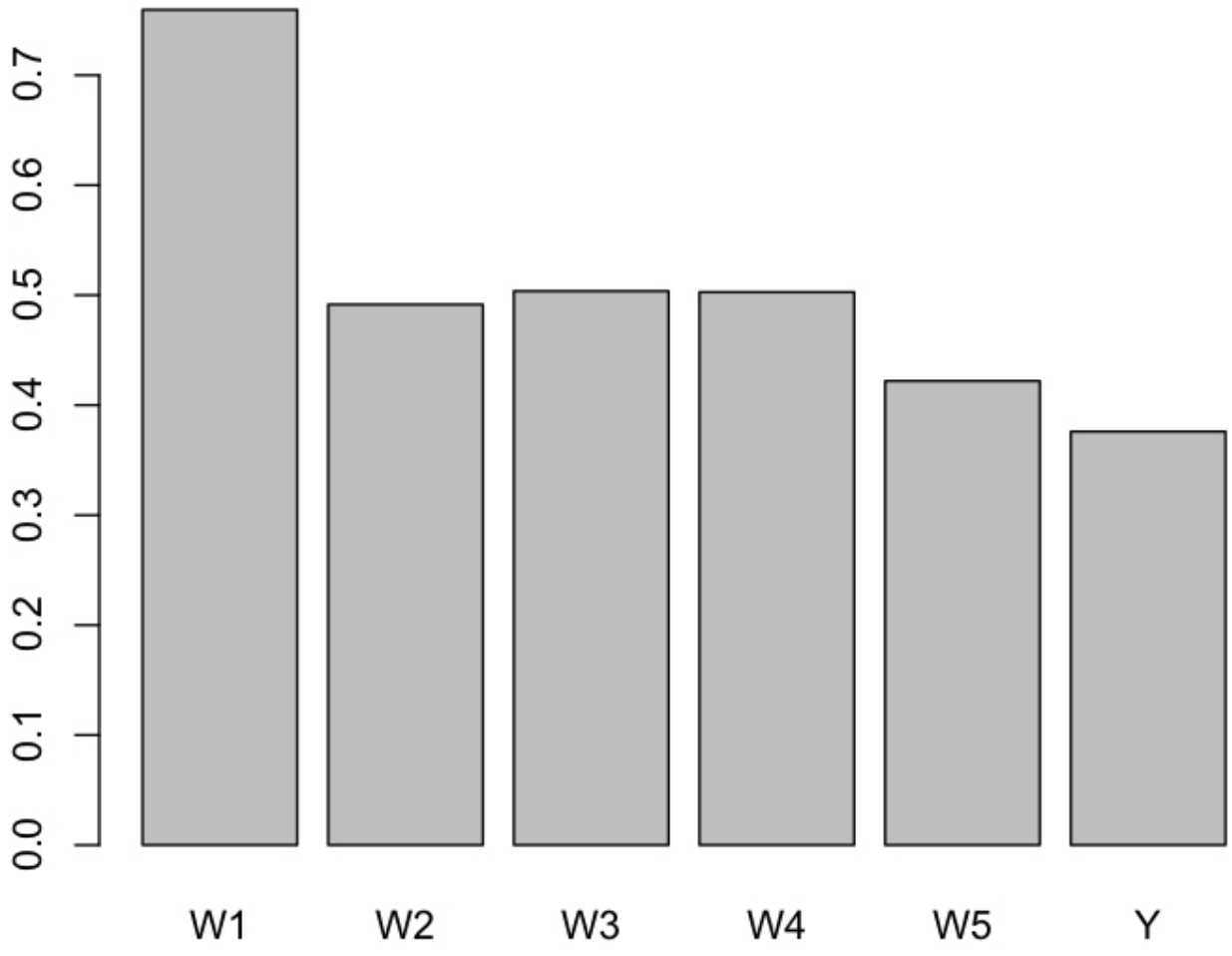
```
##Examine simulated data##

summary(data)
barplot(colMeans(data))
```

Super Learner Sample Code

```
> summary(data)
       W1                W2                W3
 Min.   :0.5007   Min.   :0.002617   Min.   :0.2517
 1st Qu.:0.6291   1st Qu.:0.240574   1st Qu.:0.3869
 Median :0.7681   Median :0.461070   Median :0.5105
 Mean   :0.7595   Mean   :0.491561   Mean   :0.5037
 3rd Qu.:0.8930   3rd Qu.:0.758717   3rd Qu.:0.6235
 Max.   :0.9996   Max.   :0.999448   Max.   :0.7494
       W4                 W5                Y
 Min.   :0.0008207   Min.   :0.000   Min.   :0.000
 1st Qu.:0.2628806   1st Qu.:0.000   1st Qu.:0.000
 Median :0.5118454   Median :0.000   Median :0.000
 Mean   :0.5027983   Mean   :0.422   Mean   :0.376
 3rd Qu.:0.7344431   3rd Qu.:1.000   3rd Qu.:1.000
 Max.   :0.9998029   Max.   :1.000   Max.   :1.000
```

Super Learner Sample Code

```
> barplot(colMeans(data))
```



Super Learner Sample Code

```
##Specify a library of algorithms##

SL.library <- c("SL.glm", "SL.mean",
       "SL.randomForest", "SL.glmnet")
```

Super Learner Sample Code

Could use various forms of "screening" to consider differing variable sets

```
SL.library <- list(c("SL.glm","screen.randomForest", "All"),
       c("SL.mean", "screen.randomForest", "All"),
       c("SL.randomForest", "screen.randomForest", "All"),
       c("SL.glmnet", "screen.randomForest","All"))
```

Or the same algorithm with different tuning parameters

```
SL.glmnet.alpha0 <- function(..., alpha=0){
       SL.glmnet(..., glmnet.alpha=alpha)}
SL.glmnet.alpha50 <- function(..., alpha=.50){
       SL.glmnet(..., glmnet.alpha=alpha)}

SL.library <- c("SL.glm","SL.glmnet", "SL.glmnet.alpha50",
       "SL.glmnet.alpha0","SL.randomForest")
```

Super Learner Sample Code

```
##Specify a library of algorithms##

SL.library <- c("SL.glm", "SL.mean",
       "SL.randomForest", "SL.glmnet")
```

Super Learner Sample Code

```
##Run the super learner to obtain predicted values for
the super learner as well as CV risk for algorithms
in the library##

set.seed(27)
fit.data.SL<-SuperLearner(Y=data[,6],X=data[,1:5],
        SL.library=SL.library, family=binomial(),
        method="method.NNLS", verbose=TRUE)
```

Super Learner Sample Code

```
CV SL.glm_All
CV SL.mean_All
CV SL.randomForest_All
CV SL.glmnet_All
Number of covariates in All is: 5
CV SL.glm_All
CV SL.mean_All
CV SL.randomForest_All
CV SL.glmnet_All
Number of covariates in All is: 5
CV SL.glm_All
CV SL.mean_All
CV SL.randomForest_All
CV SL.glmnet_All
Number of covariates in All is: 5
CV SL.glm_All
CV SL.mean_All
CV SL.randomForest_All
CV SL.glmnet_All
Number of covariates in All is: 5
CV SL.glm_All
CV SL.mean_All
CV SL.randomForest_All
CV SL.glmnet_All
Number of covariates in All is: 5
CV SL.glm_All
CV SL.mean_All
CV SL.randomForest_All
```

Super Learner Sample Code

```
> #CV risks for algorithms in the library
> fit.data.SL

Call:
SuperLearner(Y = data[, 6], X = data[, 1:5], family =
binomial(),
    SL.library = SL.library, method = "method.NNLS",
    verbose = TRUE)


                          Risk      Coef
SL.glm_All           0.1345897 0.000000
SL.mean_All          0.2353896 0.000000
SL.randomForest_All  0.1416266 0.221733
SL.glmnet_All        0.1341844 0.778267
```

Super Learner Sample Code

```
#Run the cross-validated super learner to obtain its CV risk##

set.seed(27)
fitSL.data.CV <- CV.SuperLearner(Y=data[,6],X=data[,1:5], V=10,
        SL.library=SL.library,verbose = TRUE,
        method = "method.NNLS", family = binomial())
```

Super Learner Sample Code

```
##Cross validated risks##

#CV risk for super learner
mean((data[,6]-fitSL.data.CV$SL.predict)^2)

#CV risks for algorithms in the library
fit.data.SL
```

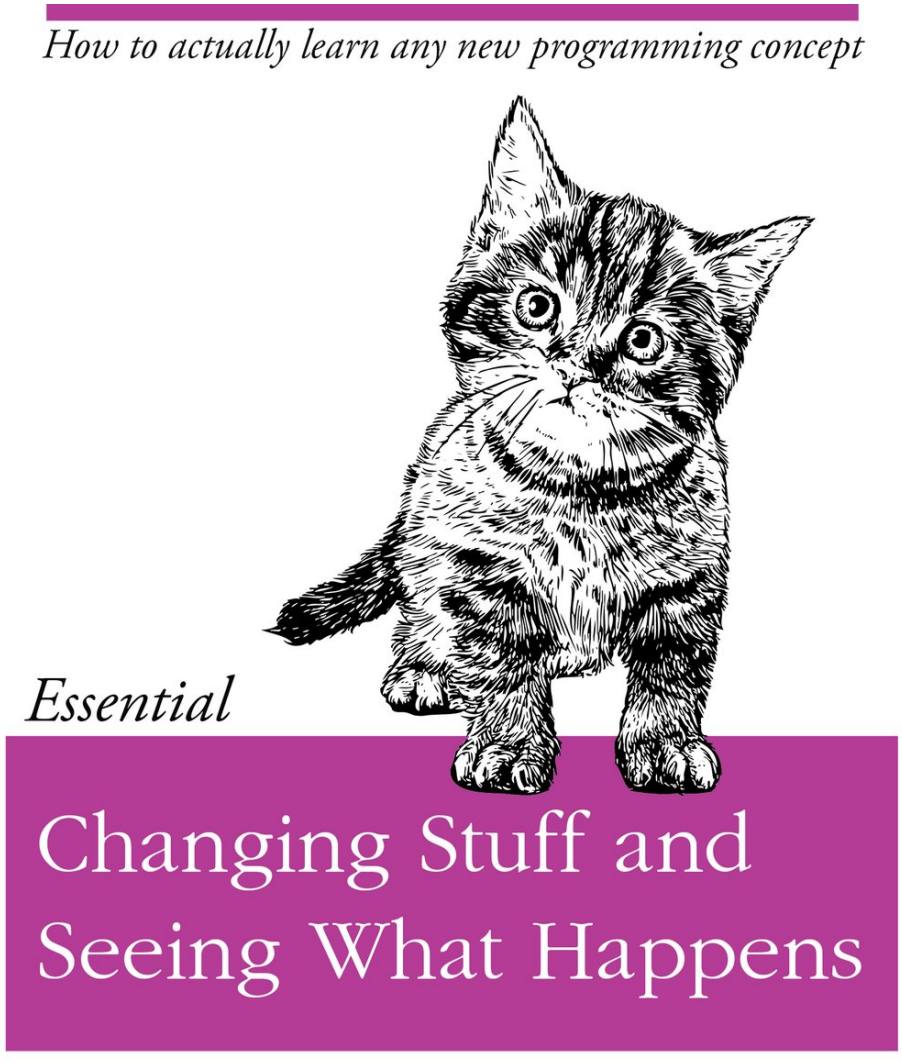
Super Learner Sample Code

```
> #CV risk for super learner
> mean((data[,6]-fitSL.data.CV$SL.predict)^2)
[1] 0.1340333

> #CV risks for algorithms in the library
> fit.data.SL

Call:
SuperLearner(Y = data[, 6], X = data[, 1:5], family =
binomial(),
    SL.library = SL.library, method = "method.NNLS",
    verbose = TRUE)


                          Risk      Coef
SL.glm_All           0.1345897 0.000000
SL.mean_All          0.2353896 0.000000
SL.randomForest_All 0.1416266 0.221733
SL.glmnet_All        0.1341844 0.778267
```

Super Learner Sample Code

```
> #CV risk for super learner
> mean((data[,6]-fitSL.data.CV$SL.predict)^2)
[1] 0.1341084
>
> #CV risks for algorithms in the library
> fit.data.SL

Call:
SuperLearner(Y = data[, 6], X = data[, 1:5], family =
binomial(),
    SL.library = SL.library, method = "method.NNLS",
    verbose = TRUE)


                          Risk      Coef
SL.glm_All            0.1345897 0.0000000
SL.glmnet_All         0.1341851 0.7769335
SL.glmnet.alpha50_All 0.1345260 0.0000000
SL.glmnet.alpha0_All  0.1344223 0.0000000
SL.randomForest_All   0.1416445 0.2230665
```

When Learning a New Package...

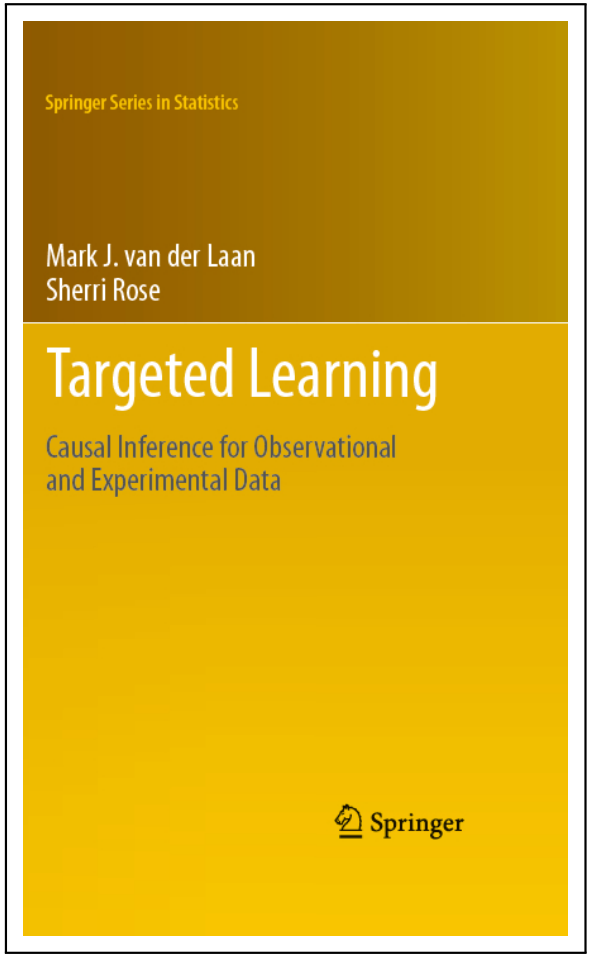


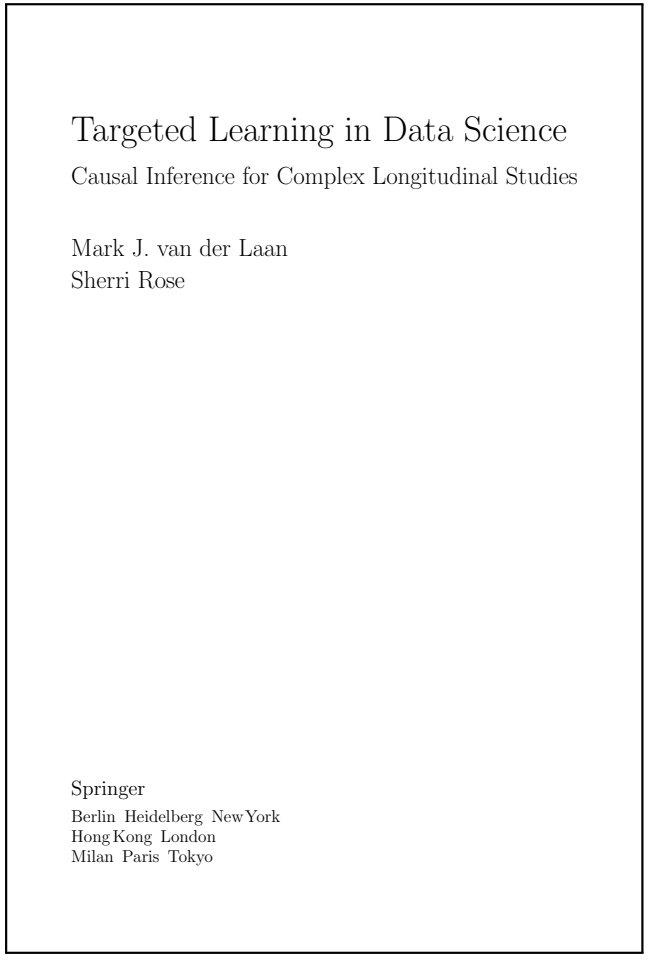
More on SuperLearner R Package

- `SuperLearner` (Polley): CRAN
- Eric Polley Github: `github.com/ecpolley`

More: `targetedlearningbook.com/software`

Targeted Learning (targetedlearningbook.com)





van der Laan & Rose, *Targeted Learning: Causal Inference for Observational and Experimental Data*. New York: Springer, 2011.

[Q & A]