Lecture 7: Introduction to Data-adaptive Estimation and Super Learning

A roadmap for causal inference

1. Specify **Causal Model** representing real background knowledge
2. Specify **Causal Question**
3. Specify **Observed Data** and link to causal model
4. **Identify** : Knowledge + data sufficient?
5. Commit to an **estimand** as close to question as possible, and a **statistical model** representing real knowledge.
6. **Estimate**
7. **Interpret** Results

Key Points- prior lecture

- Parameter: a function with input a distribution in the statistical model and output a value in the parameter space
- Estimator: a function with input a realization of the the observed data and output a value (that should be) in the parameter space

- An estimator that does not respect statistical model can lead to poor estimates
  - Some measures of estimator performance: Bias, Variance, MSE

Outline

1. Challenges of non parametric estimation
2. Introduction to data-adaptive estimation, cross validation, and loss based learning.
3. Introduction to Super Learning

References

- TLB. Chapters 1 and 3
- Polley and van der Laan. “Super Learner in Prediction” Technical Report 266, Division of Biostatistics, University of California, Berkeley, 2010. http://www.bepress.com/ucbbiostat/paper266/
- Nice talk: Le Dell www.stat.berkeley.edu/~ledell/docs/dlab_ensembles.pdf
- Simulation example: Petersen, Schwab, van der Laan- World Bank chapter on bcourses

Simulated Example: Impact of an HIV prevention intervention

- Aim to evaluate the impact of a combination prevention intervention on HIV incidence
- Target Causal Parameter: ATE
  - $E_{U,X}(Y_1-Y_0)$
- X=(W1,W2,W3,A,Y)
  - W1:baseline HIV prevalence
  - W2: existence of a trading center
  - W3: pre-intervention prevalence of circumcision
  - A=Combination prevention package
  - Y=HIV incidence
- SCM: no exclusion restrictions or independence assumptions

Simulated Example: Impact of an HIV prevention intervention

- O=X=(W1,W2,W3,A,Y)
- Observe Data from 100 communities
  - 100 i.i.d copies of $O_i=(W1_i, W2_i, W3_i, A_i, Y_i)$
  - Note: O is a community-level variable
- Statistical model: non-parametric
- Working SCM $\mathcal{M}^{\mathcal{F}^*}$such that W satisfies backdoor criteria
- Estimand: $\Psi(P_0)=E_W[E(Y|A=1,W)-E(Y|A=0,W)]$

Simulated Example: True (unknown) data generating process

- W1~N(mean=0,sd=1)
- W2~Bern(p=0.3)
- W3~N(mean=0,sd=1)
- A~Bern(p=expit(-1+0.5×$W1^2$))
  - The intervention is preferentially allocated to communities with
    - High pre-intervention prevalence of HIV (Due to perceived need)
    - Low pre-intervention prevalence of HIV (In order to leverage better infrastructure)
- Y~expit(-2.5-2×$W1^2$+0.5×W2- 0.5×A + 0.2×W3×A - 1.1×W3 + U_Y)
  - U_Y~0.2*N(mean=0,sd=1)

HIV Example: Two Simple Substitution Estimators

- Estimator 1: $E(Y|A,W)$ estimated using a correctly specified parametric regression
  - $E(Y|A,W)=$
    $\text{expit}(\beta_0+\beta_1 W1^2+\beta_2 W2+\beta_3 W3+\beta_4 A+\beta_5 A\times W3)$
  - Estimate of $\Psi(P_0)$?
- Estimator 2: $E(Y|A,W)$ estimated using a misspecified parametric regression
  - $E(Y|A,W)=\beta_0+\beta_1 W1+\beta_2 W2+\beta_3 W3+\beta_4 A$
  - Estimate of $\Psi(P_0)$?

Results: HIV Example

- Bias/variance estimated based on 500 samples each of size 100
  - Truth= -0.026

Estimator	Mean estimate	Bias	Variance
Estimator 1 (correctly specified regression)	**-0.026**	**-2e-4**	**3e-5**
Estimator 2 (misspecified regression)	**-0.043**	**-0.017**	**2e-4**

- Misspecified regression-> bias
  - Does it matter?

Confidence Interval Coverage and Type I Error Control

- Note: Both of these measures also require a variance estimator
  - We will come back to approaches to variance estimation
- Confidence Interval Coverage
  - In what proportion of repetitions of the identical experiment do the constructed confidence intervals contain the true parameter value?
  - Ideally, a 95% CI contains the truth in 95% of repetitions

Confidence Interval Coverage and Type I Error Control

- Type I Error Control
  - Type I error: falsely rejecting the null hypothesis
    - The Investigator concludes an effect is present when it is not
  - In a simulation in which the null hypothesis is true, in what proportion of repetitions of the identical experiment is the null hypothesis rejected?
  - Ideally, using a 5% significance level should result in the null being falsely rejected in 5% of repetitions

Results: HIV Example

- Estimated based on 500 samples of size 100

Estimator	95% CI Coverage	Type I Error
Simulation 1: Truth=-0.026		
Estimator 1 (correctly specified regression)	95%	NA
Estimator 2 (misspecified regression)	68.2%	NA
Simulation 2: Truth= 0.00		
Estimator 1 (correctly specified regression)	95%	5%
Estimator 2 (misspecified regression)	66.8%	33.2%

Misspecified parametric regression

- If your true statistical model is non-parametric, reliance on misspecified parametric regression models can lead to

1. Biased point estimates
2. Misleading statistical inference

-> Misleading conclusions

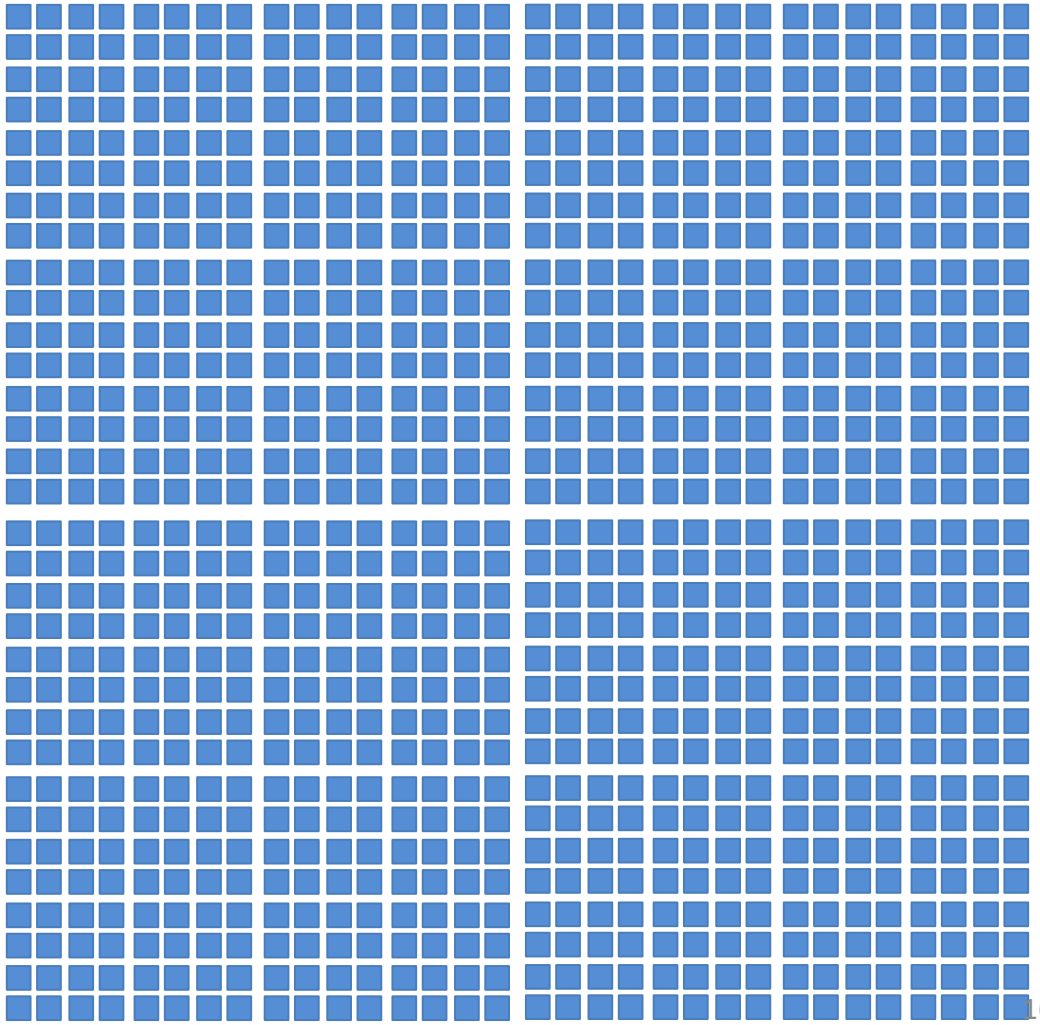
- This bias does not decrease with increasing sample size
  - With sample size big enough, you will always reject the null hypothesis, even when it is true

Estimation in a non-parametric statistical model

- If (A,W) low dimensional and we have enough subjects- could estimate the mean of Y separately in each stratum of (A,W)
  - Estimate coefficients in a saturated regression model
- Number of parameters (# of coefficients) needed for this approach grows exponentially with dimension of A,W
  - Quickly get into a situation where # parameters > # subjects
  - Even if we don't, a fully saturated model may be an overfit of the data- we return to this in a moment

Number of strata increases exponentially with dimension of W!

# of binary RV's in W	# of strata of W
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024



Estimation in a non-parametric model

- Why not look at the data?
  - Don't make any *a priori* assumptions, just see which estimator works best
- This is in fact what we do… but be careful!
  - An estimator must be an *a priori* specified algorithm
  - If not, can introduce bias and misleading inference

Dangers of looking at the data in an *ad hoc* way…

- Example: try a bunch of regression specifications, look at the results, confer…

1. Bias
   - End up picking a model specification that gives you the answer that makes the most sense to you… (or your collaborators)
   - Even if the null hypothesis is true, if this procedure is repeated over and over, it can on average lead to rejecting the null
     - Particularly (but not only) with big sample size

Dangers of looking at the data in an ad hoc way…

- Example: try a bunch of regression specifications, look at the results, confer…

2. Misleading assessment of uncertainty in your estimate (ie variance of your estimator)
   - Your confidence interval/p-value estimates are based on assumption that the model specification was *a priori* specified
   - If you ignore that you tried several models, there is more uncertainty in the process than you are acknowledging

This doesn't mean we can't look at the data, we just need to do it in a rigorous ("supervised") way…

- OK to look at multiple candidate estimators of E(Y|A,W) but…

1. Need to specify the candidates ahead of time
2. Need a rigorous, automated, pre-specified way way to choose between candidates

- With these ingredients, our estimator <u>includes</u> the selection process
  - Remember: Our estimator is just a function that takes as input the observed data and gives a number (an *estimate* ψ_n of the *estimand* ψ_0) as output

Data-adaptive estimation

- Automated algorithms for learning from data
  - While respecting the statistical model
- Computer Science: Supervised machine learning
  - Terms used interchangeably here
- Today: a brief conceptual introduction to a very big topic
- Focused on basics of Loss-based learning and V-fold cross validation

HIV Example Continued

- We don't know enough to specify a priori a correct parametric model for E(Y|A,W)
- We could write down a set of possible candidate parametric regressions. For example:
  - Logistic, main terms only
    - $E(Y|A,W)=expit(\beta_0+\beta_1 W1+\beta_2 W2+\beta_3 W3+\beta_4 A)$
  - Logistic, some interactions with A
    - $E(Y|A,W)=expit(\beta_0+\beta_1 W1+\beta_2 W2+\beta_3 W3+\beta_4 A+\beta_5 A*W3)$
    - $E(Y|A,W)=expit(\beta_0+\beta_1 W1+\beta_2 W2+\beta_3 W3+\beta_4 A+\beta_5 A\times W1+\beta_6 A\times W2+\beta_7 A\times W3)$
  - Logistic, all possible interactions (~W1*W2*W3*A)
    - $E(Y|A,W)=expit(\beta_0+\beta_1 W1+\beta_2 W2+\beta_3 W3+\beta_4 A+\beta_5 A\times W1+\beta_6 A\times W2+\beta_7 A\times W3+\beta_8 W1\times W2+\beta_9 W2\times W3+\beta_{10} W1\times W3+\beta_{11} W1\times W2*A+\beta_{12} W2\times W3\times+\beta_{13} W1\times W3\times A+\beta_{14} W1\times W2\times W3+\beta_{15} W1\times W2\times W3\times A)$
  - Lots more possible candidates
    - Ex. Linear versions of above
    - Ex. Polynomials (eg $W1^2$, $W3^2$, $W1^3$,...)

Choosing Between Candidates

- Some of these candidates will fit the data better than others…
- Bias-Variance tradeoff
  - An estimator with too few parameters (too little complexity) will be overly biased
    - Example: simple linear model for a highly non-linear process
  - An estimator with too many parameters (too much complexity) will be overly variable
    - Example: One parameter per observation

Overfitting

- Too much complexity- Responding too much to noise (detail) in training data
  - Example: same number of parameters as number of observations
    - Can predict Y in current sample perfectly
    - Will not do a good job on a different sample from the same distribution
    - In other words- variance is too high
- Can't just fit a bunch of regressions using all the data and choose the one that does the best job predicting Y in the same data
  - Various solutions to this problem
  - We will focus on one… Loss-based estimation

How to evaluate our candidate estimators of $E_0(Y|A,W)$?

- We want the best estimator of $E_0(Y|A,W)$
  - Our goal is to estimate the entire function

$$\bar{Q}_0 : (A, W) \rightarrow \bar{Q}_0(A, W)$$

  - Our estimator must
  1. Take as input the observed data: n i.i.d. copies of O=(W,A,Y)
  2. Give as output a prediction function that maps any (A,W) into a "predicted value" for (i.e. estimated expectation of) Y
- We need to define what we mean by "best"
  - **Loss function** provides a measure of performance

Loss Functions

- Loss function applied to observation O assigns a measure of performance to a candidate function for $E_0(Y|A,W) \equiv \bar{Q}_0$
  - In other words, it is a a function of Random variable O and candidate $\bar{Q}$

$$L : (O, \bar{Q}) \rightarrow L(O, \bar{Q}) \in \mathbb{R}$$

- Example: L_2 (Squared Error) Loss Function

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

- Example: Negative log loss function

$$L(O, \bar{Q}) = -log(\bar{Q}(A,W)^Y (1 - \bar{Q}(A,W))^{1-Y})$$

Loss-Based Learning in a Nutshell

1. Define target $\bar{Q}_0$ as the minimizer of the expectation of a loss function (or "Risk")

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

2. Generate an estimate of the risk for each candidate $\bar{Q}$
3. Choose the candidate with the smallest estimated risk

– Assuming we can estimate risk well, this gives us the candidate closest to the true target (with respect to the measure of dissimilarity implied by the loss function)

- Difference in risk at the candidate and optimal risk at $\bar{Q}_0$

Back to the ATE

- Our target is $E_0(Y|A,W) \equiv \bar{Q}_0$
- We want a loss function such that

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

  - Expectation under P_0 minimized by $E_0(Y|A,W)$
- True for L2 loss function

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

  - For binary Y, also true for –log loss function

$$L(O, \bar{Q}) = -log(\bar{Q}(A,W)^Y (1 - \bar{Q}(A,W))^{1-Y})$$

    - If Y is a continuous RV with outcomes in [0,1], this is no longer the –log likelihood loss function, but it is still a valid loss function for $E_0(Y|A,W)$

Big picture

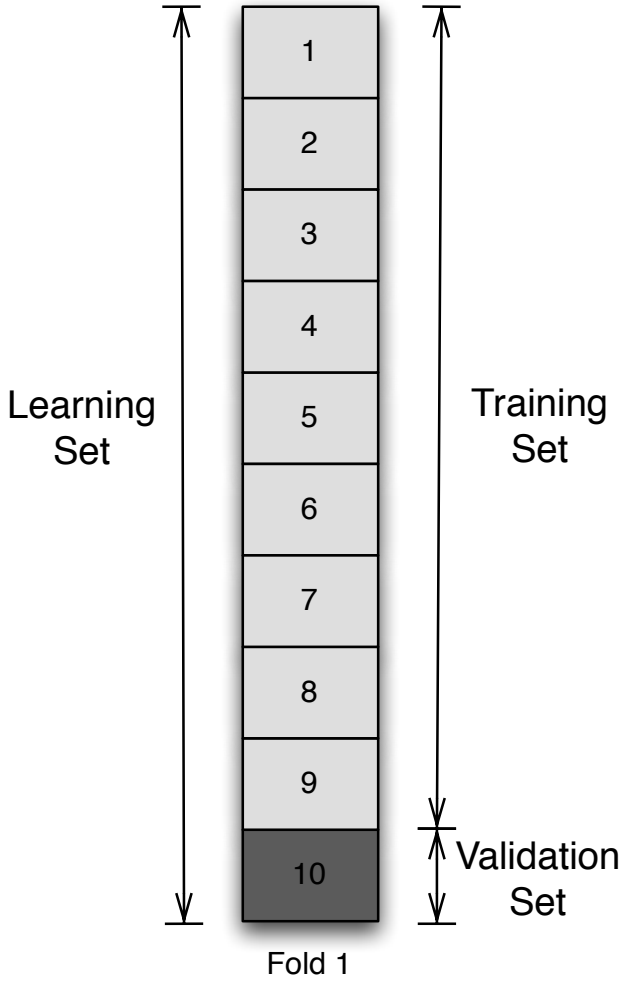
- We now have a way to quantify the relative performance of different candidate estimators of $E_0(Y|A,W)$
- We want the candidate that gives the smallest expected value of the loss function (or risk)
  - Ex: For L_2 loss function, the candidate with the smallest mean squared prediction error
  - This makes intuitive sense:
    - MSE=bias^2 +variance
    - We want an estimator with small bias and variance
- We still need a way to estimate the risk…
  - One we have a risk estimate for each candidate, we can use it to choose the best candidate

Cross-Validation: Big picture

- Allows us to compare algorithms based on how they perform on independent data from the same distribution
  - When building the candidate prediction functions, reserve a piece of the data (called the validation set)
  - Use the validation set to compare the performance of the candidate prediction functions fit by the competing algorithms
    - Eg based on mean squared prediction error
- Lots of types of cross validation
  - We will focus on one: V-fold

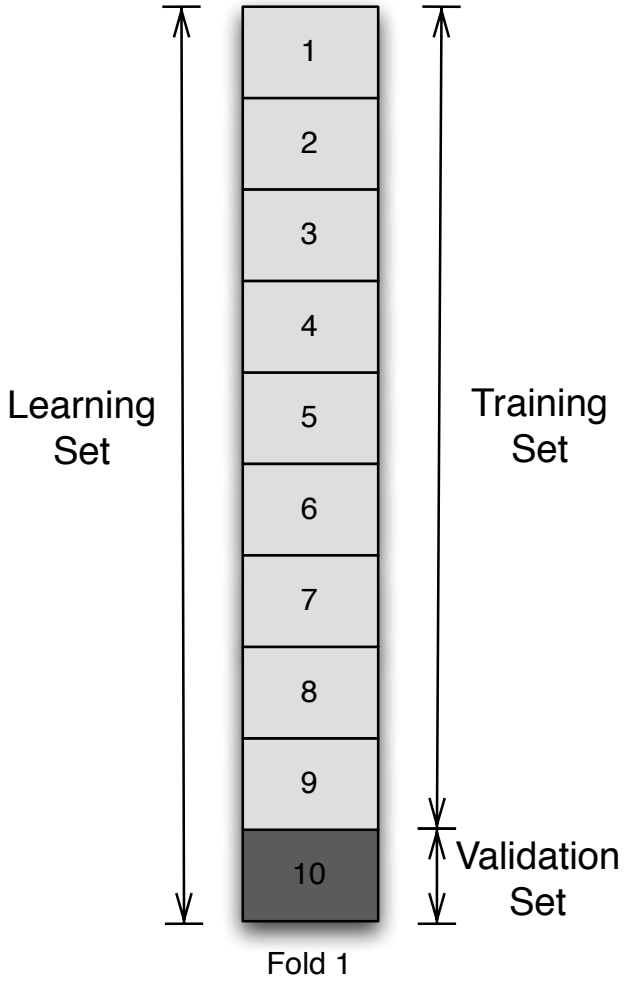
V-fold Cross-Validation

- Observed data $O_1,\ldots,O_n$ is the learning set
- We partition the learning set into V sets of size≈n/V
  - Here V=10
- For a given fold, one set is the validation set and the remaining V-1 are the training set



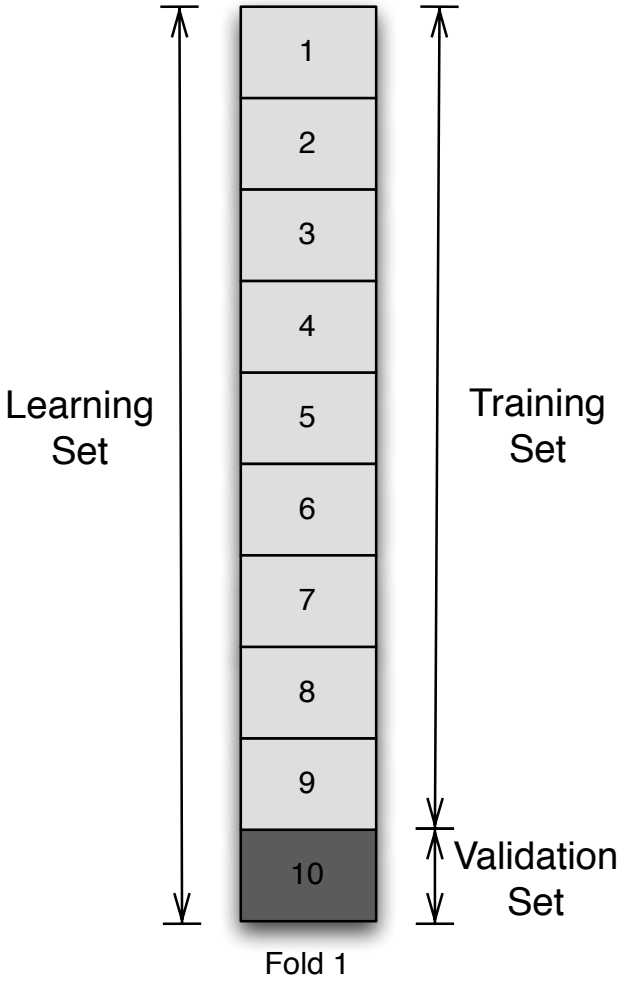
V-fold Cross-Validation

- Observations in the training set are used to construct (or train) the candidate estimators
- For example, we fit each of our candidate parametric regressions only using data from the training set



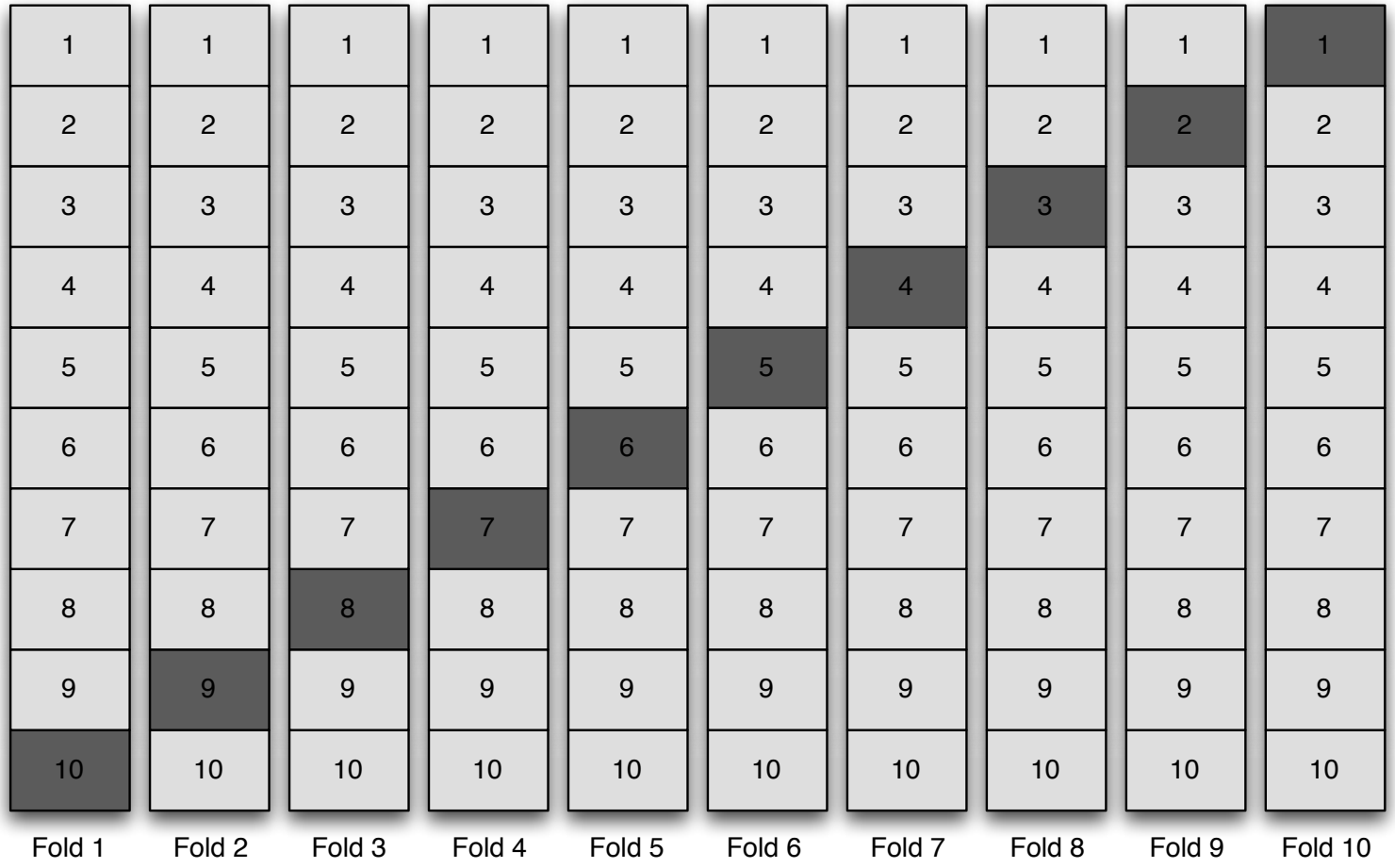
V-fold Cross-Validation

- The observations in the Validation set are used to assess the performance (estimate the risk) of the candidate estimators
- For example, we calculate how well (eg in terms of MSE) each candidate regression (fit on the training set) does at predicting the outcome in validation set



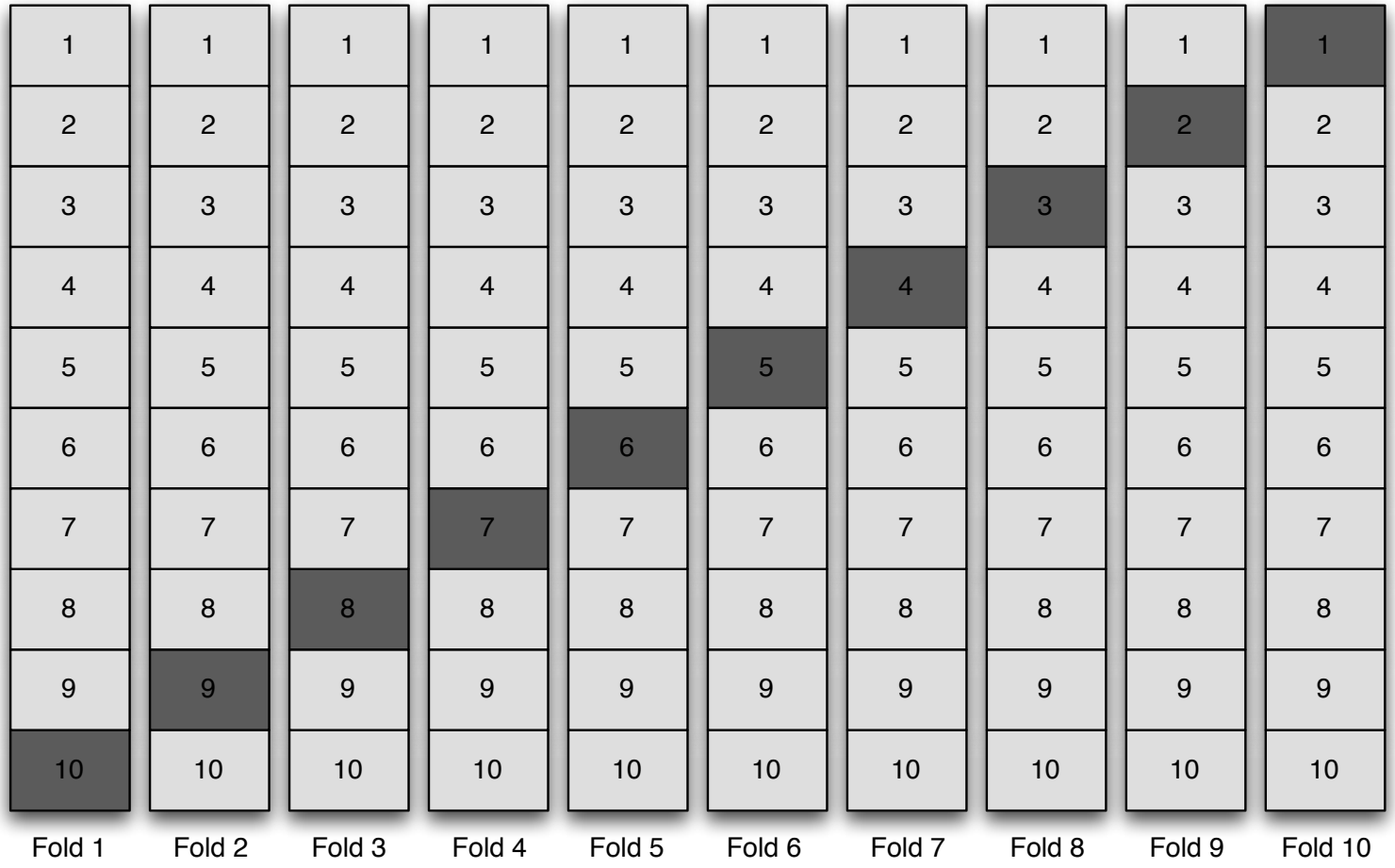
V-fold Cross-Validation

- The validation set rotates V times such that each set is used as the validation set once.



V-fold Cross-Validation

- Risk estimated as average of risk (eg squared prediction error) estimated in each validation set



Expanding the library of candidate algorithms

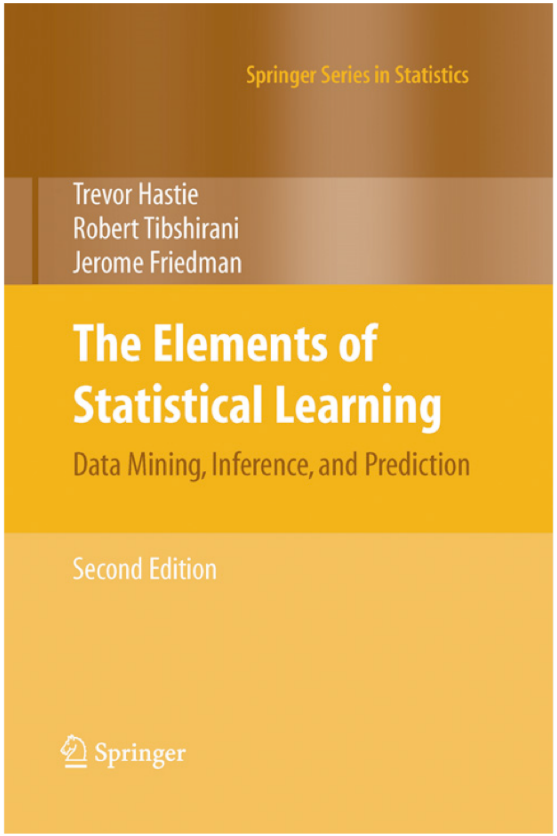
- When selecting candidates, we don't have to limit ourselves to parametric regression models
  - All we need for a given candidate is that it takes as input our observed data and gives as output a prediction function
- Lots of fancier approaches are out there
  - Many of these approaches are themselves data-adaptive algorithms
  - Ie. They look at the data in a supervised way in order to build a predictor
  - For example, they may themselves do cross-validation
- We refer to our *a priori specified* set of candidate algorithms as our library

Lots of data adaptive algorithms!

A few examples

- Forward or backward stepwise selection
- K nearest neighbor
- Multiple Additive Regression Splines (MARS)
- CART, Random Forests
- Neural networks
- Least Angle Regression (LARS)
- Polynomial spline regression
- …

One reference (and great book)



The R Library of Prediction Algorithms

- The key is a good library of machine learning algorithms
- Currently >100 R packages for machine learning/prediction

`http://cran.r-project.org/web/views/MachineLearning.html`

- Can expand library further by using, eg
  - Different candidate covariates (features)/dimension reductions
    - Different approaches to screening
  - Parametric regression models based on background knowledge

Which algorithm to choose?

- Each of these algorithms might work wonderfully for some prediction problems and terribly for others
  - Ex- Parametric model is great if correctly specified, can be (but will not necessarily be) terrible if not…
- It is very difficult (impossible?) to know which one will work best for a given problem
  - Background knowledge can give us an idea of algorithms that might work well, but we may be wrong
- Why not choose the algorithm that performs best for the current prediction problem?

The Dangers of Favoritism

- Relative cross validated Risk (compared to main term regression “least squares”)

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

Van der Laan et al; Super Learner (2007) SAGMB;6(1)

Overview of Super Learning

- Ensemble approach: “stacked generalization”
- Set up a competition between algorithms
- Specify

1. Which candidate algorithms get to compete
   - We refer to our *a priori specified* set of candidate algorithms as our library
2. How you will judge the winner
   - Choose a loss function
     - Ex: Squared error (L2) for E(Y|A,W)
   - Estimate risk (expectation of the loss function) using V-fold cross-validation

- Apply the winning algorithm to the full dataset

Discrete Super Learner (or the Cross Validation Selector)

- Choose the algorithm that gives us the best predictor for our specific prediction problem and data
  - Based on estimated Risk (using cross validation)
- We will do as well asymptotically (or very close) as the best algorithm in our library
  - Oracle results for cross-validated loss-based learning
    - Loss function must be bounded
  - Also get good finite sample behavior

Summary : Discrete SL

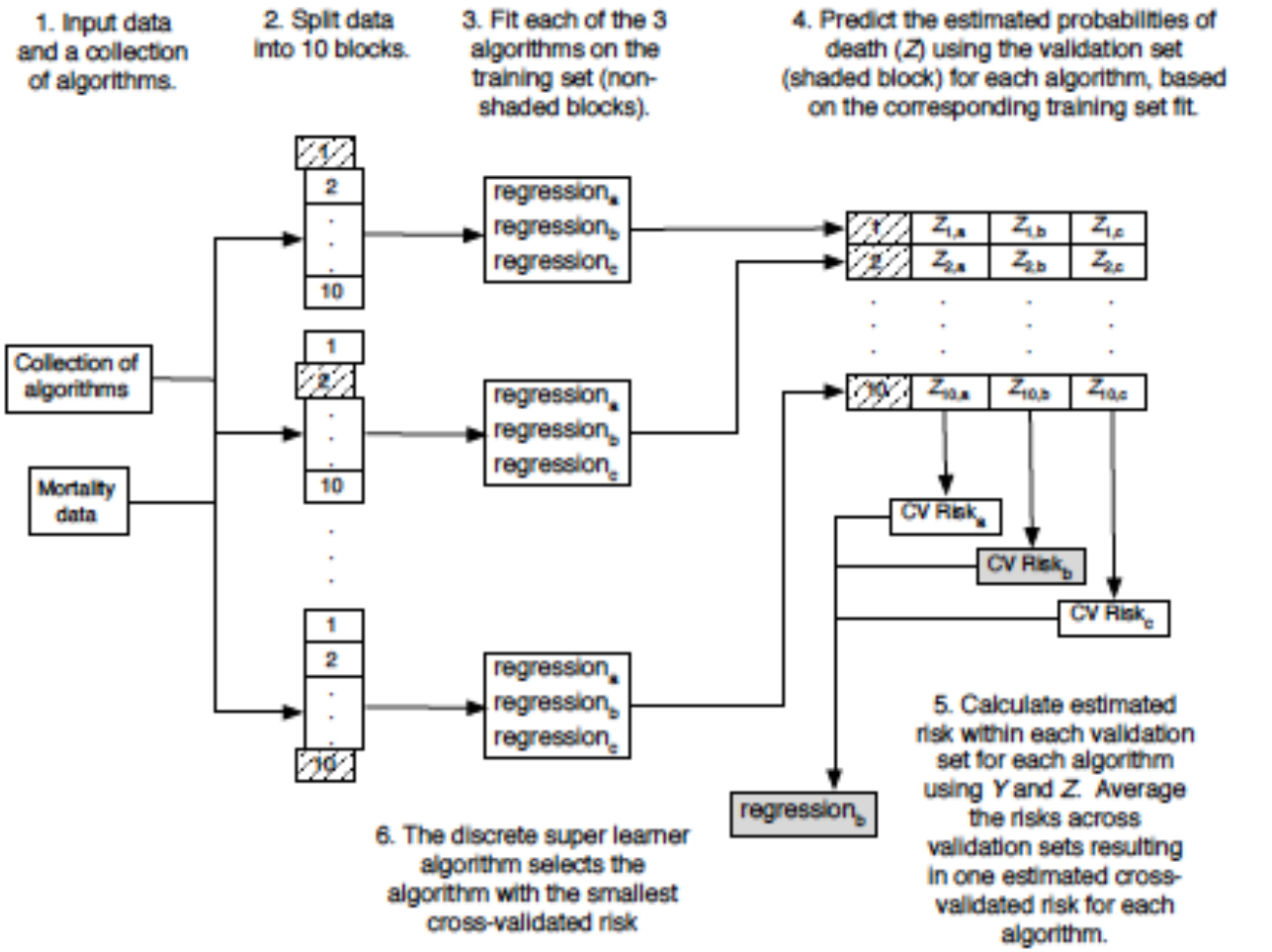


Fig. 3.1 Discrete super learner algorithm for the mortality study example where $Q_n^b(A, W)$ is the algorithm with the smallest cross-validated risk

TLB Chapter 3

Discrete Super Learner (or the Cross Validation Selector)

- Selects the algorithm with lowest estimated risk (best performance on validation data), and reruns on full data for final prediction model

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

Van der Laan et al; Super Learner (2007) SAGMB;6(1)

Back to our HIV Example…

- We have a set of possible candidate parametric regressions (all logistic).
  - #1 Correctly specified
    - Includes $W1^2$ term and interaction W3 and A
  - #2 Main terms
  - #3 Logistic, all possible interactions (~W1*W2*W3*A)
  - #4 Interactions of A with W1, W2, W3
- We use 10-fold cross validation to estimate the risk of each
  - Using the L2 loss function

HIV Example: Estimated Risk

- How would you use Cross Validation to estimate the risk of each candidate (L2 loss)?
- Which candidate would discrete SL choose?

Candidate	Estimated Risk
#1 (correctly specified)	1.37e-3
#2 (main terms only)	1.98e-2
#3 (all interactions)	1.37e-1
#4 (all two way interactions with A)	2.01e-2

Beating the Best Algorithm

- The discrete Super Learner can only do as well as the best candidate in our library
- Not Bad, but,
- We can do even better…

Ensemble Super Learner

- We can expand the library by also including all the weighted averages of the initial candidates
  - Ie.- Let the initial set of candidate algorithms "team up"
- Each weighted average is a unique candidate in our expanded library
  - One (or more) of these weighted combinations might perform better than any of the initial algorithms alone
  - *Or not…* Each initial algorithm also remains a candidate

Ensemble Super Learner

- We now want to choose the “best” weighted average of all of the candidates
  - “Best”: weighted average that minimizes the risk
- Cross-validation guides the selection of the optimal weighted combination
- Ensemble SL is no more computer intensive than discrete SL
  - i.e. once you have fit each of your initial candidates, selection of the optimal weight vector is computationally trivial

Finding the optimal weighted combination

- Once the discrete Super Learner has been completed
  - Each of the candidate algorithms fit on each training set
1. Choose family of weighted combinations of the initial library of algorithms, indexed by weight vector α.
   - Consider only α -vectors that sum to one, where each weight is non-negative
2. Determine which weighted combination minimizes the cross-validated risk
   - Regress the outcome Y on the cross-validated predicted values of each of the algorithms (Z_a,…Z_p).
   - Ex: for L2 Loss:

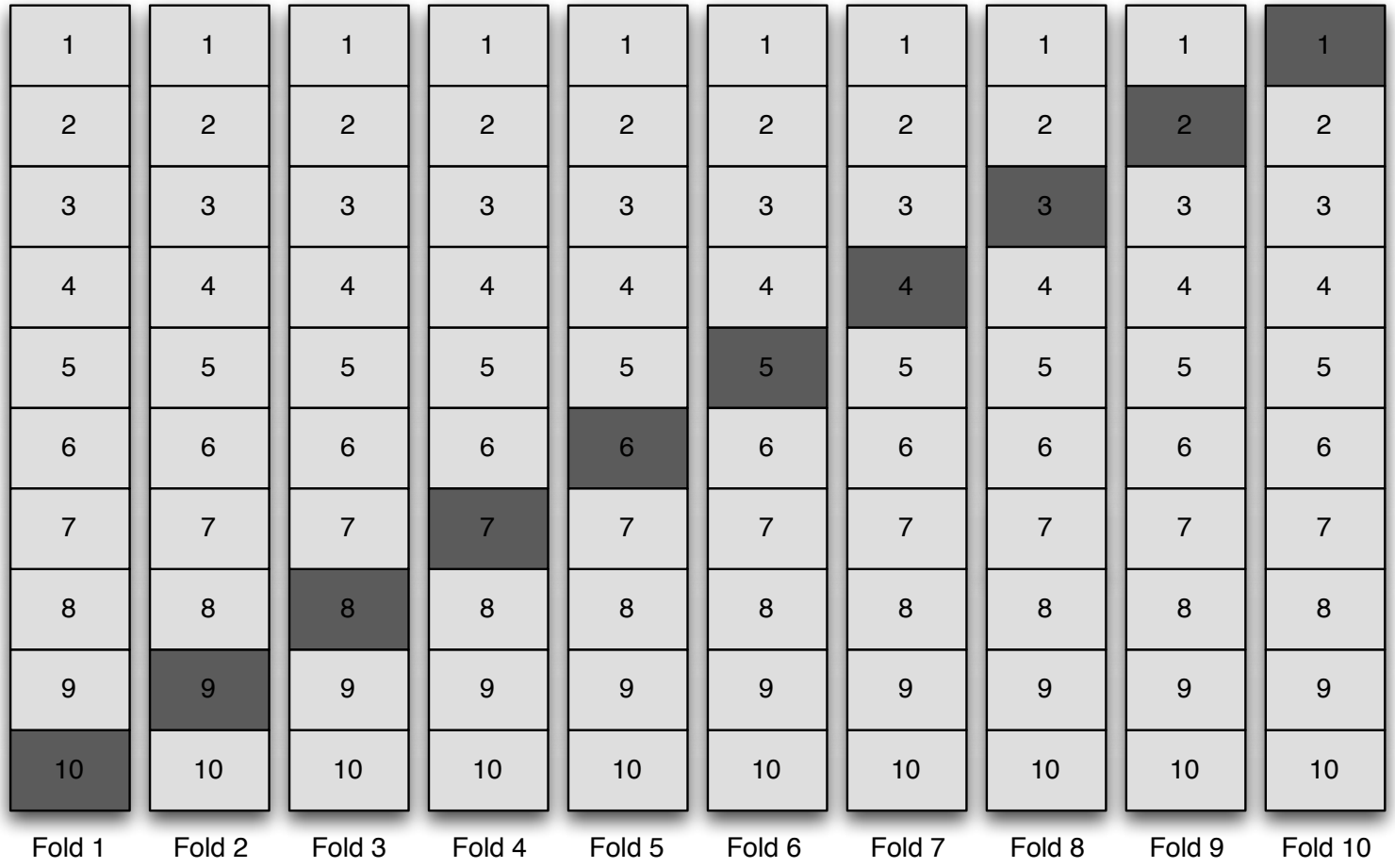
$$E_n(Y|Z) = \alpha_{a,n}Z_a + \alpha_{b,n}Z_b + \ldots + \alpha_{p,n}Z_p$$

Finding the optimal weighted combination

- Regress the Y on the cross-validated predicted values of each of the algorithms (Z_a,…Z_p).
  - L2 Ex: $E_n(Y|Z) = \alpha_{a,n}Z_a + \alpha_{b,n}Z_b + \ldots + \alpha_{p,n}Z_p$
  - For each observation i=1,…,n, regress
    - Outcome Y_i: observed outcome *on*
    - Predictors Z_i: predicted outcome for that subject (i.e. given that subject's covariates Wi) according to each algorithm (a,…,p) fit on the corresponding training set
- This fits the weight vector α that minimizes the cross-validated MSE (i.e. cv risk estimate)
  - Algorithms that predict the outcome well will get larger coefficients (bigger weights)

Recall: V-fold Cross-Validation

- Risk estimated as average of risk (eg squared prediction error) estimated in each validation set



Evaluating the performance of the SL

- Super Learner is a data-adaptive algorithm
  - The process we have outlined so far uses the whole learning set to build a prediction function
- We might want to go one step further and evaluate the performance of Super Learner
  - To check against overfitting
  - To compare to other algorithms
- The same principle applies- when evaluating performance we want to use data that SL didn't get to look at when building a prediction function
  - i.e. we want an "honest" estimate of the Risk

Evaluating the performance of the SL

- Solution: An additional layer of cross validation
  1. Partition the data into V folds
  2. Run the whole SL algorithm in each training set
     - Thus each training set will itself be partitioned into V folds in order to run SL
     - Some of the algorithms in the SL library may themselves use a third layer of cross validation….
  3. Evaluate performance on the corresponding validation sets

Finite sample performance

Four simulated datasets ($n = 100$)

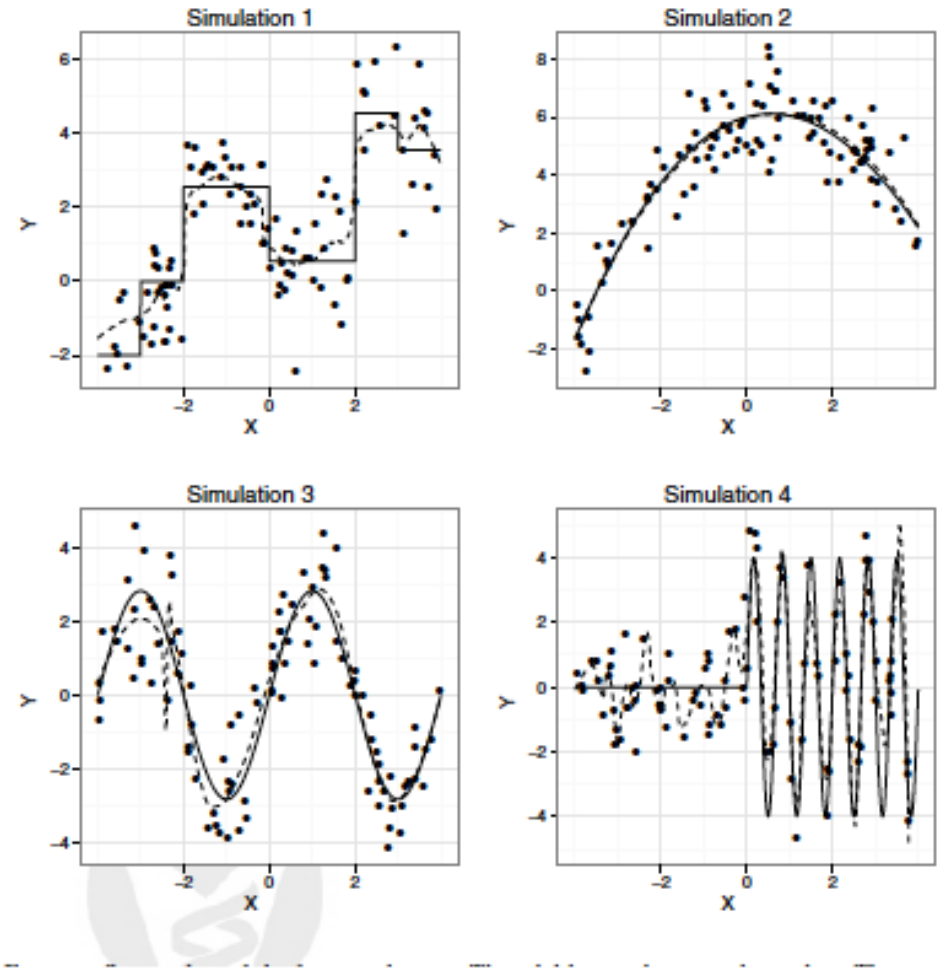


Fig. 2, Polley and van der Laan, 2010

Super Learner: Simulated Data

- Estimated cross validated risk (Mean Squared Error) relative to least squares

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 Learner	0.02	0.67	0.16	0.22

Van der Laan et al; Super Learner (2007) SAGMB;6(1)

Super Learner: Real Data

- Super Learner, applied to 13 publically available datasets

TABLE 4

Description of data sets. 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]

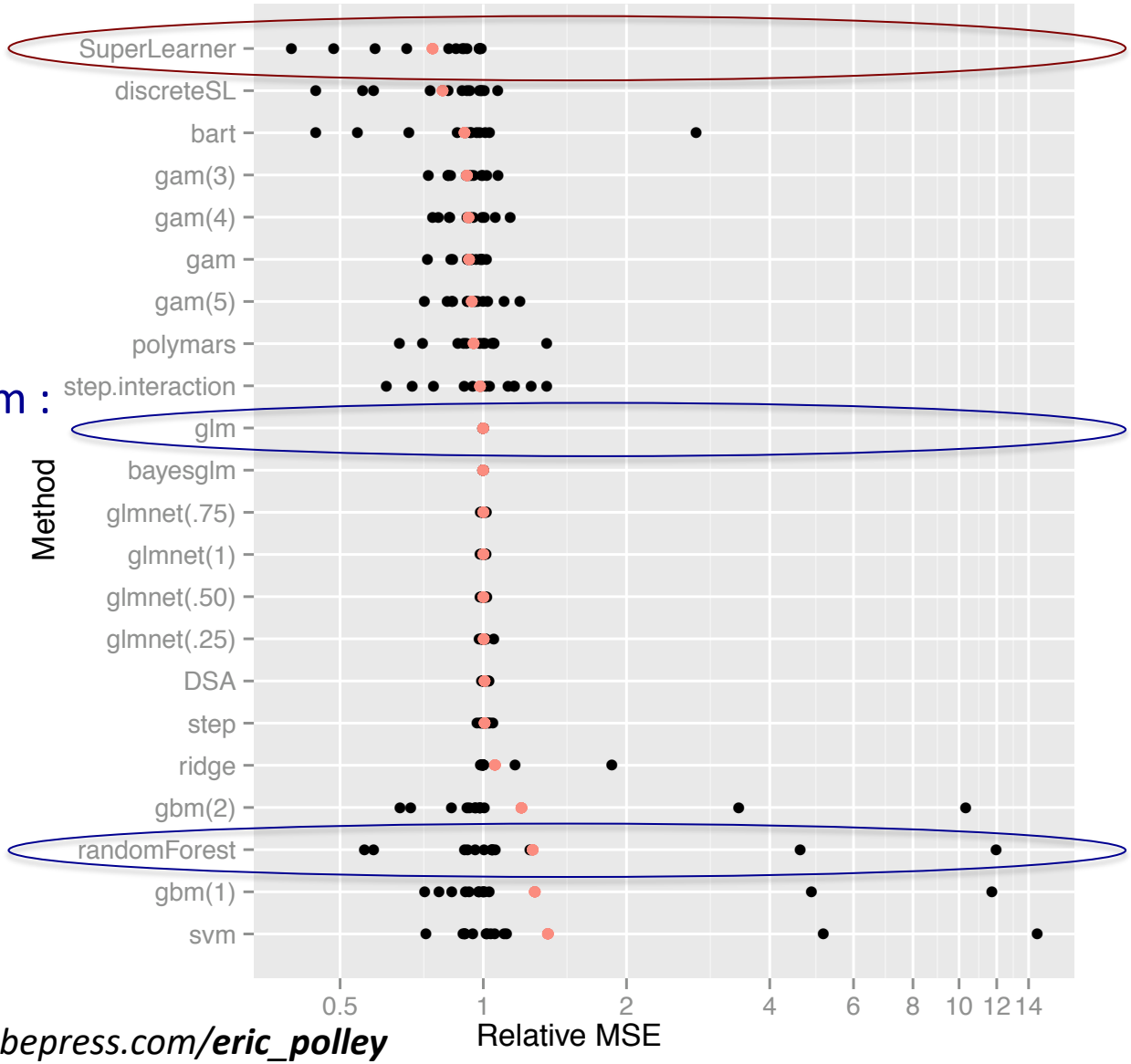
Technical Report: *works.bepress.com/**eric_polley***

Super Learner: Real Data

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

Example algorithm : Linear Main Term Regression

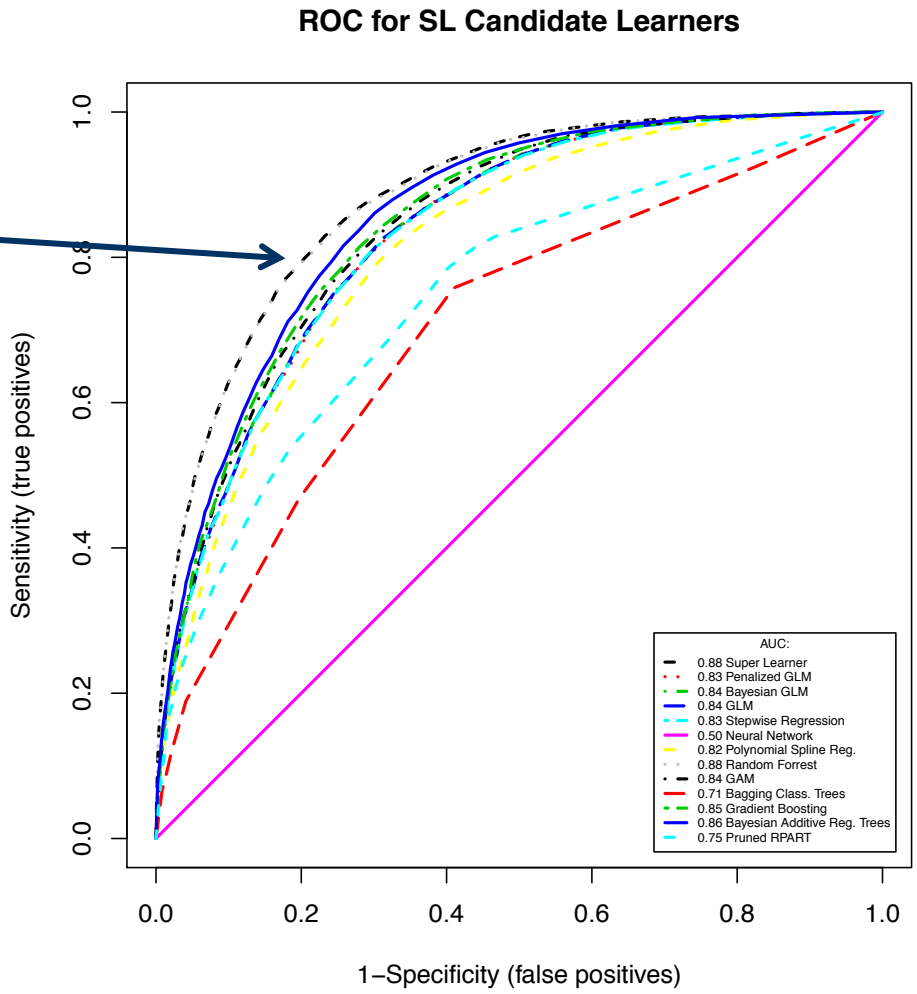
Example algorithm: Random Forest



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

Super Learner in practice: Better mortality prediction for ICU patients

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



Pirracchio, Lancet Resp Med 2015

Super Learner in practice: Better mortality prediction for ICU patients

Cross-validated Area under the Receiver-Operating Curve

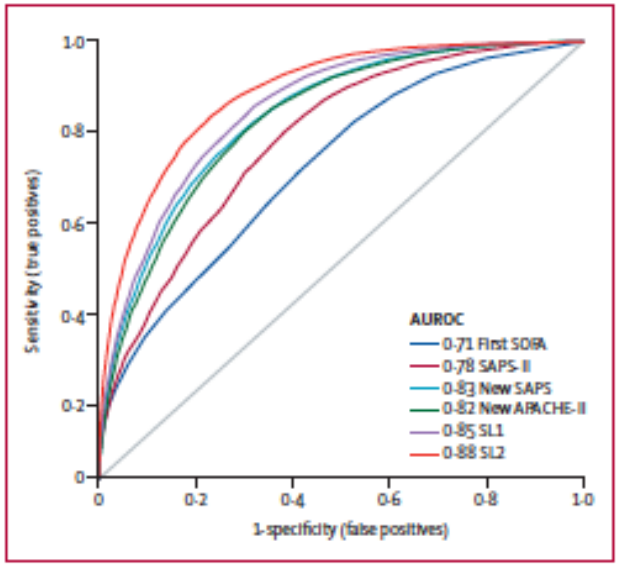


Figure 1: Receiver-operating characteristics curves

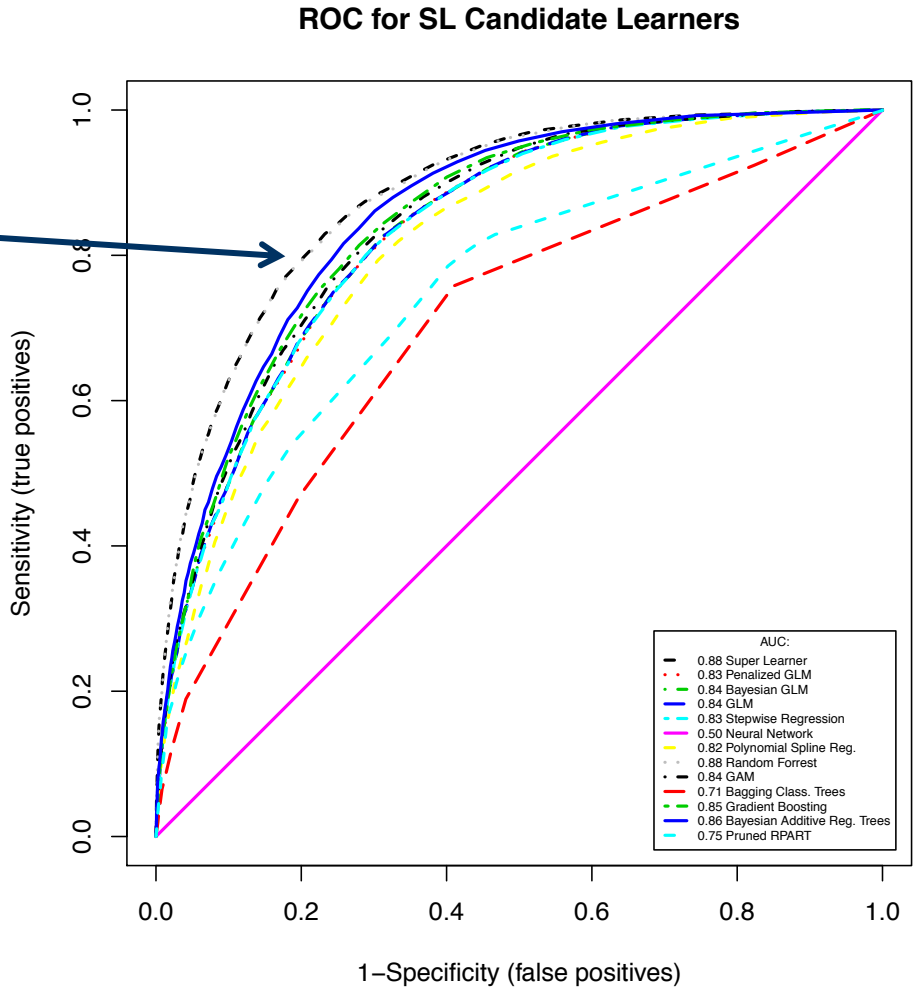
Pirracchio, et al, *Lancet*, 2014

- Sepsis-related Organ Failure Assessment (SOFA)
- Simplified Acute Physiology Score (SAPS-II)
- Acute Physiology and Chronic Health Evalution (APACHE)
- Super Learner, standard categorized variables (SL1)
- Super Learner, non-transformed variables (SL2)

- SL better distinguishes between high and low risk patients

Ensemble Super Learner can do better than any of the initial candidates alone

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

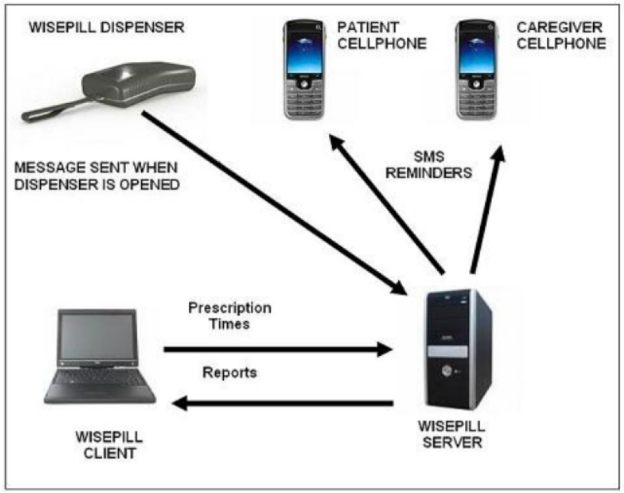


Pirracchio, Lancet Resp Med 2015

Super Learner in Practice Remote Electronic Adherence Monitoring

- Mobile technologies can

1. Monitor pill container openings remotely
   - Medication Event Monitoring System (MEMS)
2. Transmit data in real time over the cellular network



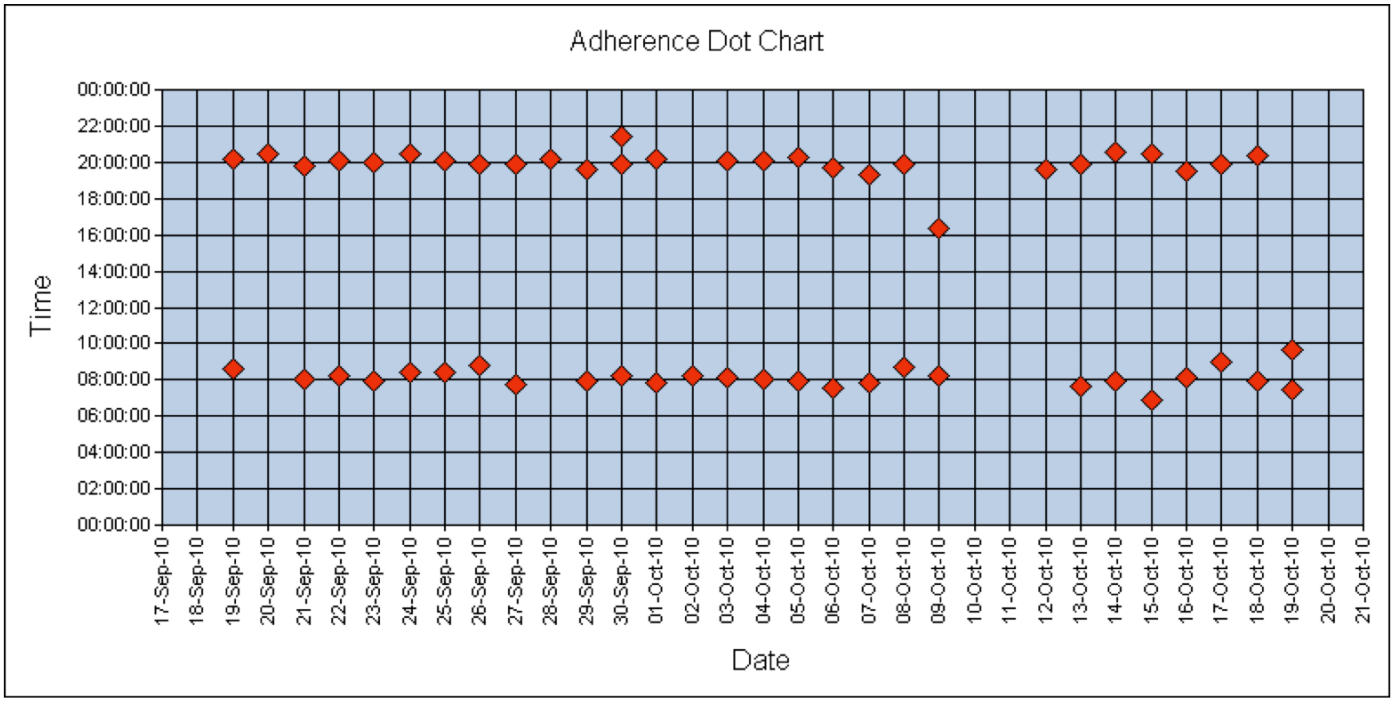
Haberer et al, 2010

Real-Time Adherence Monitoring

- Personalized monitoring schedule?
  - *Target viral load testing to patients at higher risk of virologic failure?*
- Personalized clinic visit schedule?
  - *Trigger clinic visits and adherence interventions based on adherence?*
- Detect adherence problems and intervene to prevent virologic failure?

Can Electronic Adherence Data Predict Virologic Failure?

- MEMS data: multiple events over long periods
- MEMS measures adherence imperfectly
- Adherence-> failure relationship is complex



How to build the best predictor? (*Too many variables, too little knowledge*)

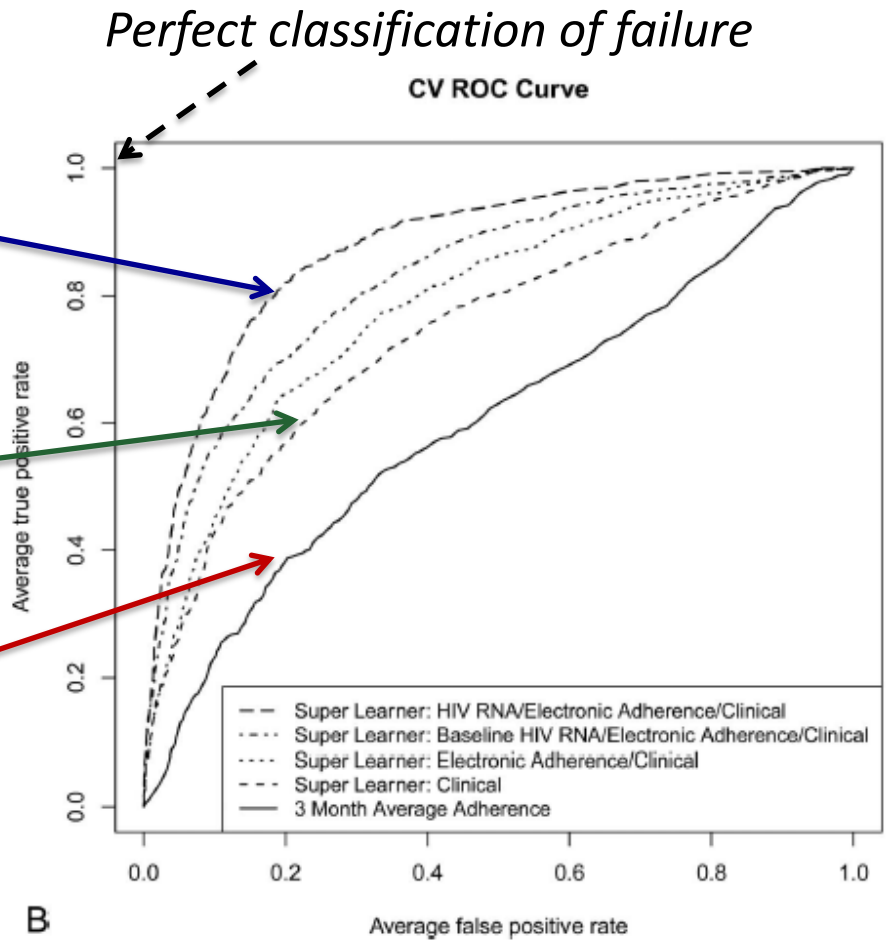
- Single variable?
  - Example: Average daily adherence over past 90 days
  - Give up a lot of information
- Logistic regression?
  - Which covariates to include?
  - Non-linear relations? Interactions?
- Machine-learning : Automated approaches to flexibly discover complex relationships from data

Improved Risk Prediction/Classification

Super Learner with MEMS and Clinical Predictors/HIV RNA

Super Learner with clinic variables only:

Average 3 month adherence

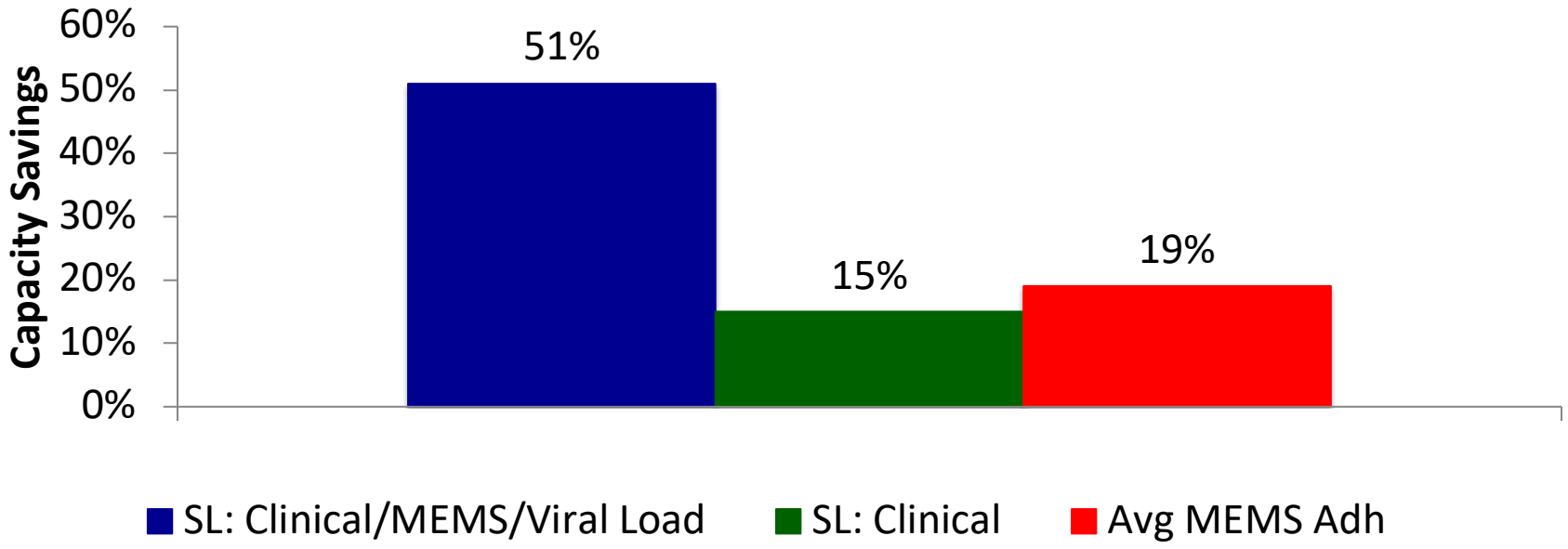


- 1478 patients with HIV treated with ART in US, 1997-2009

Petersen et al, 2015

Reduced Viral Load Testing Frequency?

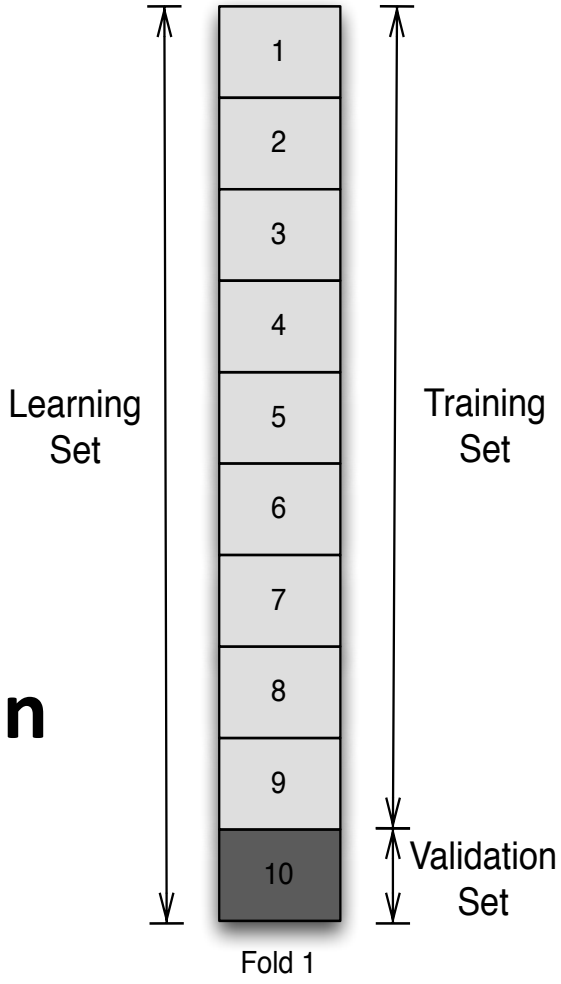
- Test only when predicted risk of failure > cutoff
  - ➢ Choose cutoff so <5% of failures have delayed detection
- Capacity savings: % of tests potentially avoided



Petersen et al, 2015

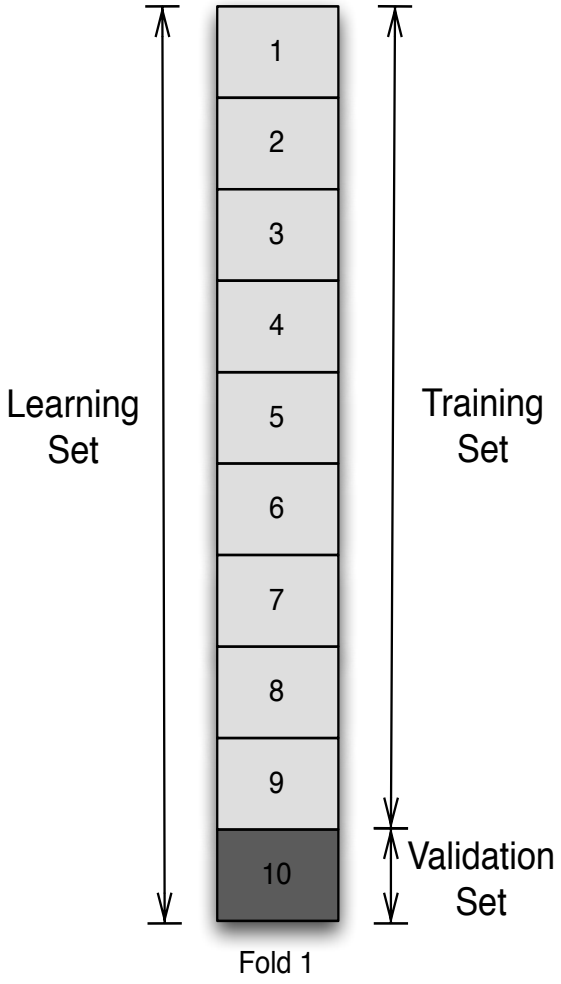
Super Learner with repeated measures

- Or other types of hierarchical/clustered data
  - For example, have for each individual (W(t),Y(t)), t=1,…,K
- Training and Validations sets should be independent
- **Use individual not record when splitting into folds**



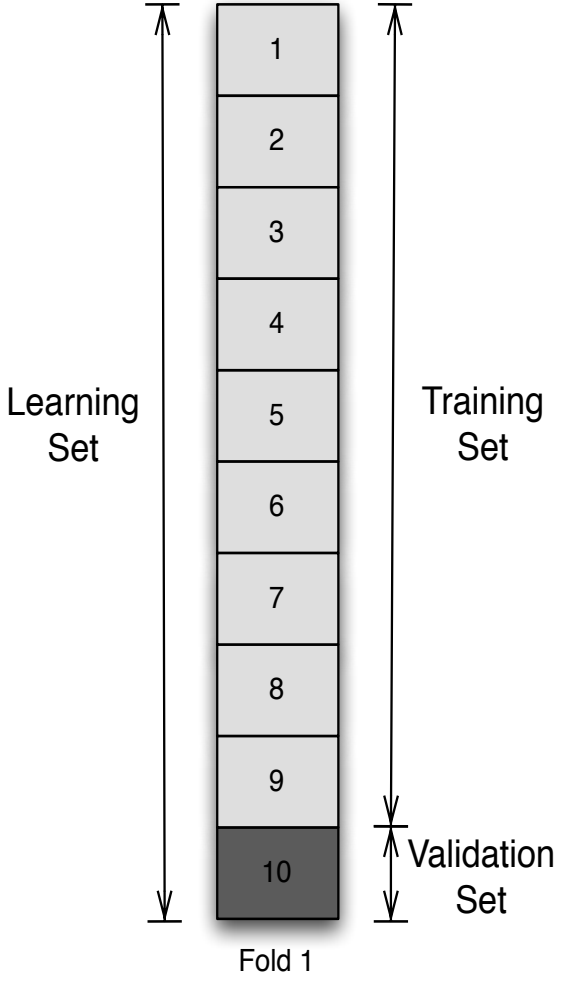
Super Learner with rare binary outcomes

1. Stratify by the outcome (Y=1 and Y=0), split into folds separately, and then combine
   - Preserves the outcome frequency in each validation set
   - Prevents validations sets with no variability in the outcome



Super Learner with rare binary outcomes

2. Case control sampling
   - Can improve computation time (and sometime performance)
   - Keep all observations with Y=1 and known fraction with Y=0
   - Use known sampling weights



Key points

- Use of an estimator that does not respect the statistical model can result in bias, and misleading inference
- Defining a good non-parametric estimator can be difficult
  - NPMLE breaks down in most realistic settings
  - We want to look at the data and pick the estimator that does best
  - If we do not treat this “looking” as part of our estimator, we run into trouble

Key points

- Super learning: choose the estimator that performs best for your data/problem

1. Choose a loss function- a measure of performance
   - Ex: Squared error or negative log
2. Measure performance fairly
   - Cross validation lets you evaluate performance using data the estimator did not get to see

Key points

- Build a big library of candidate algorithms
  - Can include data adaptive algorithms and parametric regressions
- Discrete Super Learner: Choose the algorithm with the lowest cross validated risk
  - Ex. lowest cross validated MSE
- Super Learner: Choose the convex combination of algorithms with the lowest cross validated risk
- Additional layer of cross validation to evaluate the performance of Super Learner

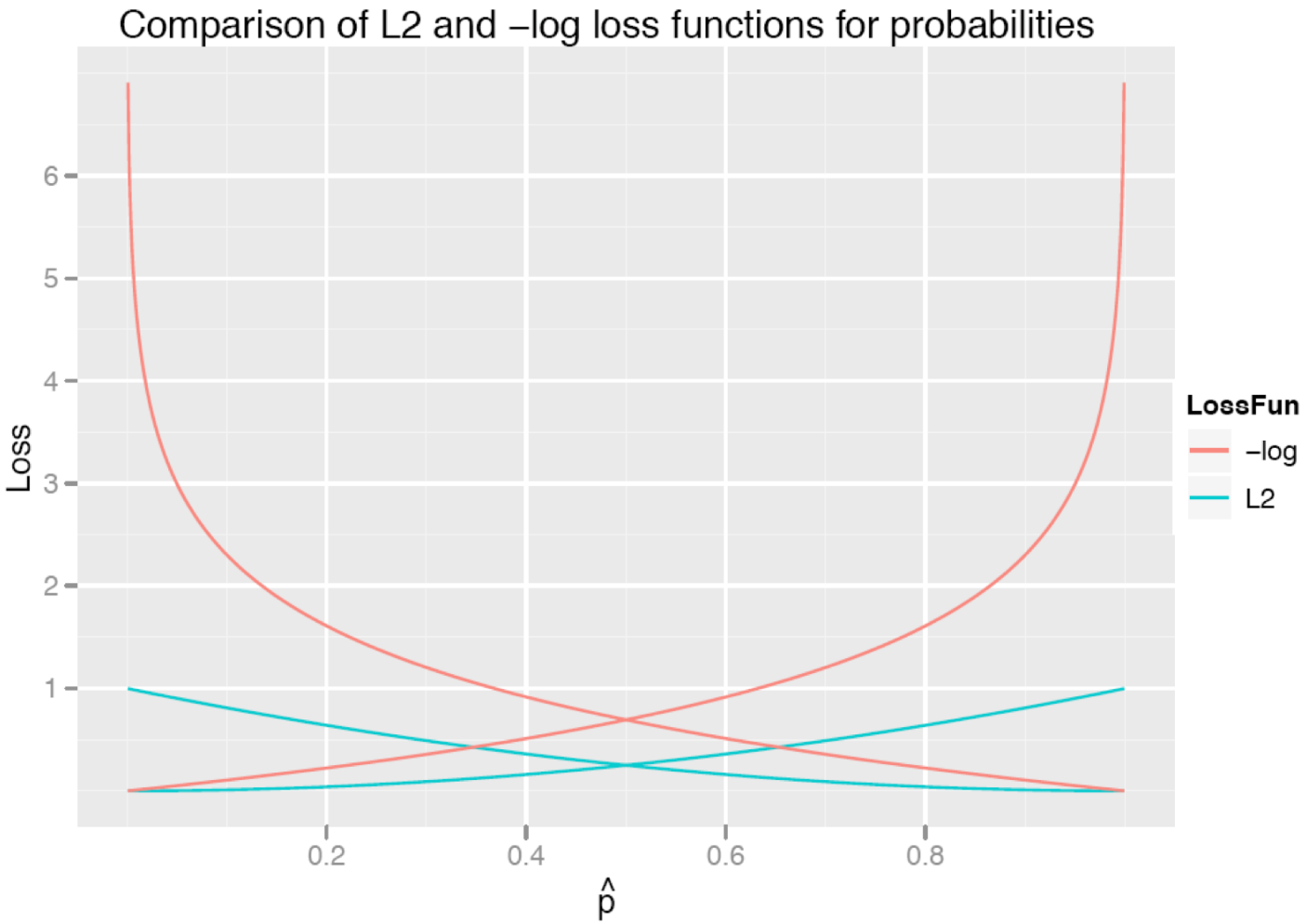
Extra

Summary: Oracle Results

- The Oracle selector is the best estimator among the K algorithms in the SL library
  - Chooses the algorithm whose fit on the training samples yields the smallest risk (expected loss) under P_0
  - Unknown: depends on both observed data and P_0
- Discrete super learner performs as well as the Oracle selector, up to a second order term.
  - Assuming a bounded loss function
  - Number of algorithms in the library polynomial in sample size

See p. 64-65, TLB; van der Laan and Dudoit 2003 for precise Oracle inequality

Loss Function Must be Bounded



Super Learner Algorithm History

- 1992) David H. Wolpert: “Stacked Generalization” in the context of neural net ensembles. Used leave-one-out CV to generate level-one data.
- (1996) Leo Breiman: “Stacked Regressions” extended Wolpert’s stacking framework to regression problems. Proposed k-fold CV to generate level-one data. Suggested non-negativity constraints for the metalearner.
- (1996) Michael Leblanc, Rob Tibshirani: “Combining Estimates in Regression and Classification” provided a general framework for stacking and compared CV-generated level-one data to bootstrapped level-one data.
- (2007) Mark van der Laan, Eric Polley, Alan Hubbard: “Super Learner” provided the theory for stacking. Guarantees asymptotic equivalence to the oracle.

Le Dell 2015: https://www.stat.berkeley.edu/~ledell/docs/dlab_ensembles.pdf

Super Learner Flow Chart

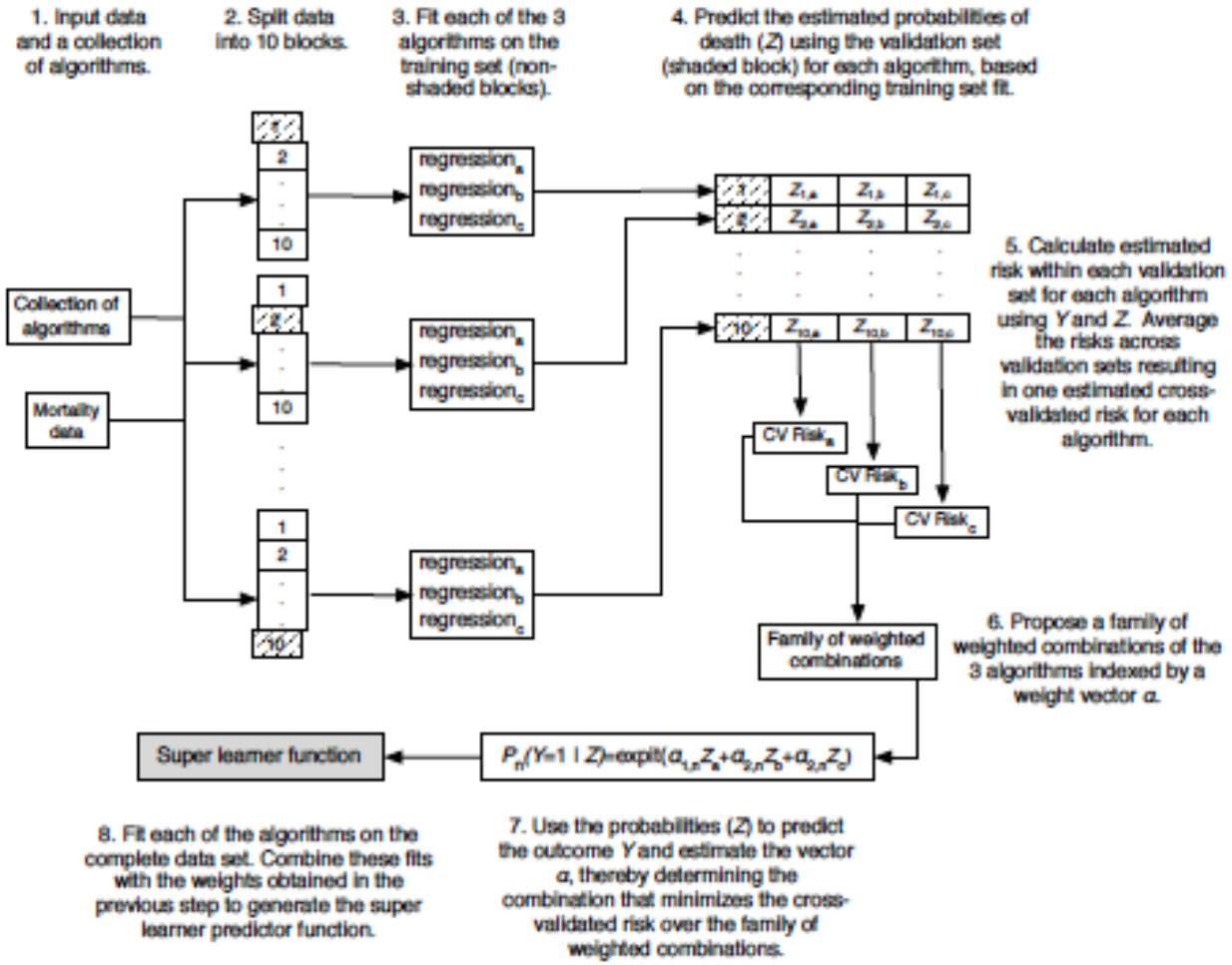


Fig. 3.2 Super learner algorithm for the mortality study example

TLB Chapter 3