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

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/

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 conclusions
3. Misleading statistical inference

- Note that bias does not decrease with increasing sample size
  - With sample size big enough, a biased estimator will lead you to 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)
  - i.e. Fit a saturated model
- Number of parameters (# of coefficients) needed 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

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…
     - MC Boily: "Evaluation Pressure"
   - Even if the null hypothesis is true, if this procedure was repeated over and over, it can on average lead to rejecting the null, even with huge 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 includes the selection process
  - Remember: Our estimator is just a function that takes as input the observed data and gives a number (estimate of the estimand) as output

Data adaptive estimation

- Automated algorithms for learning from data
  - While respecting the statistical model
- In computer science: machine learning
  - Terms used interchangeably
- We will give a conceptual overview of a very big topic
- Focus on Loss-based learning and V-fold cross validation

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: saturated model that results in sparse cells

Overfitting

- Estimator has too much complexity
  - Example: same number of parameters as number of observations
    - Predicts Y in current sample perfectly, but 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(Y|A,W)?

- We want the best estimator of E(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
    - Take as input the observed data
    - Gives as output a prediction function that maps any (A,W) into a predicted value for 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)

Back to the ATE

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

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

Big picture

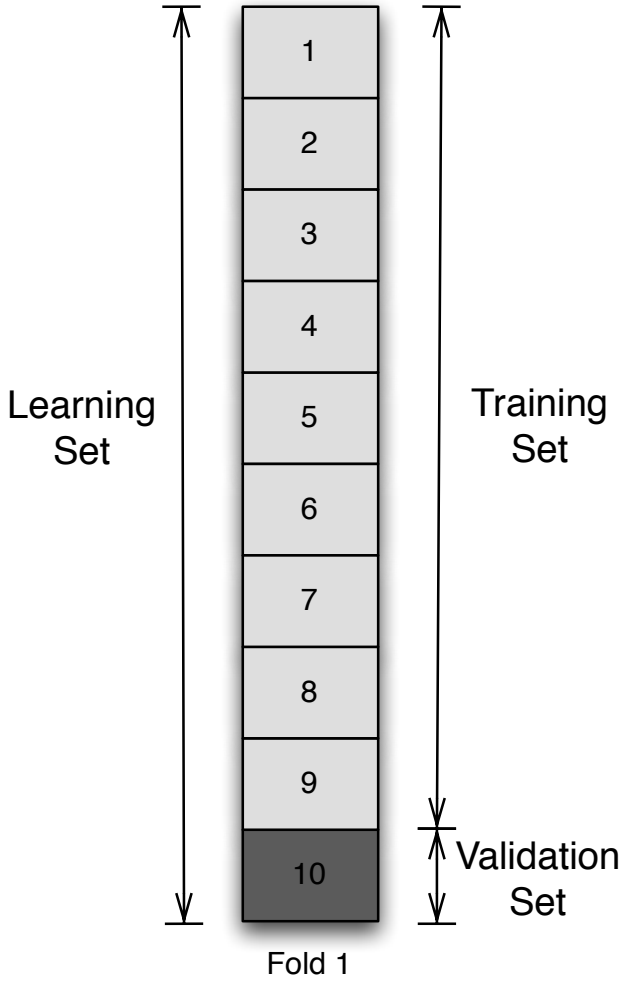
- We now have a way to quantify the relative performance of different candidate estimators of $E_0(Y|A,W)$
- We want the candidate that gives the smallest expected value of the loss function (or risk)
  - For L_2 loss function, the smallest mean squared prediction error
  - This makes intuitive sense: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

Cross-Validation: Big picture

- Allows us to compare algorithms based on how they perform on independent data from the same distribution
  - When building the predictive models, reserve a piece of the data (called the validation set)
  - Use the validation set to compare the performance of the prediction models 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

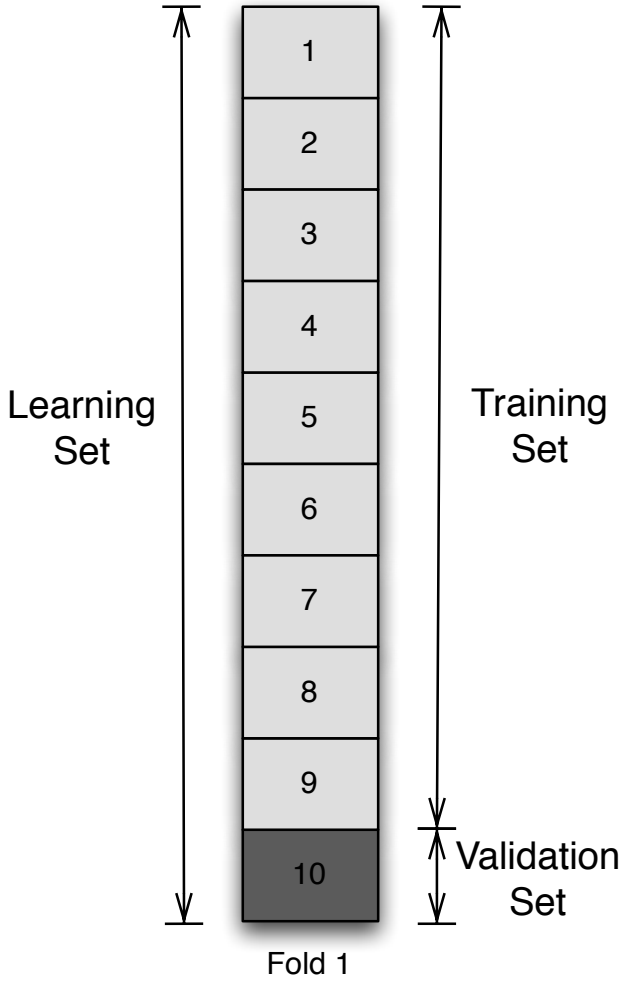
V-fold Cross-Validation

- 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



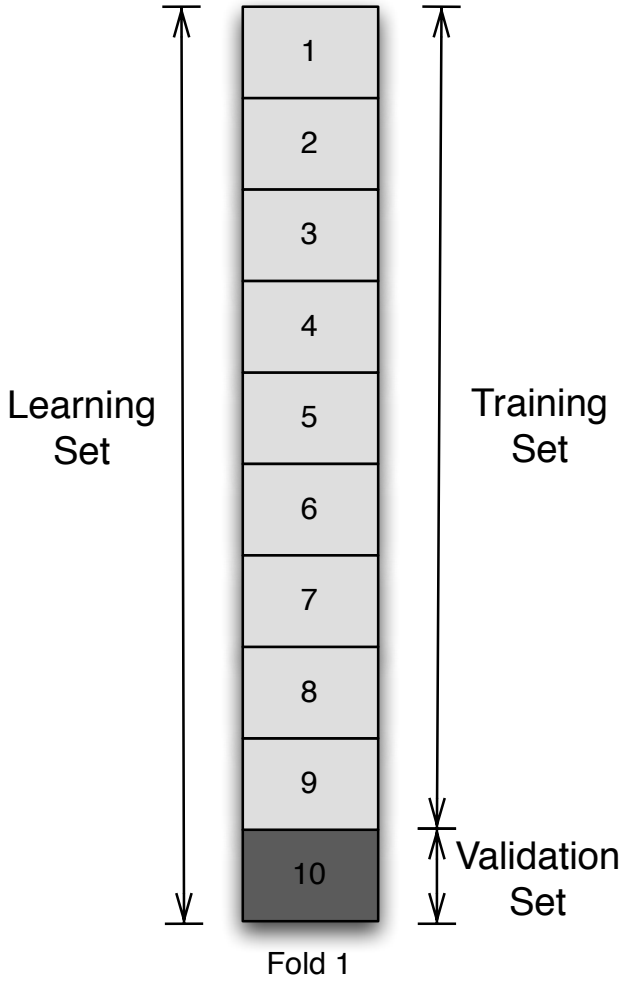
V-fold Cross-Validation

- 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 for the training set



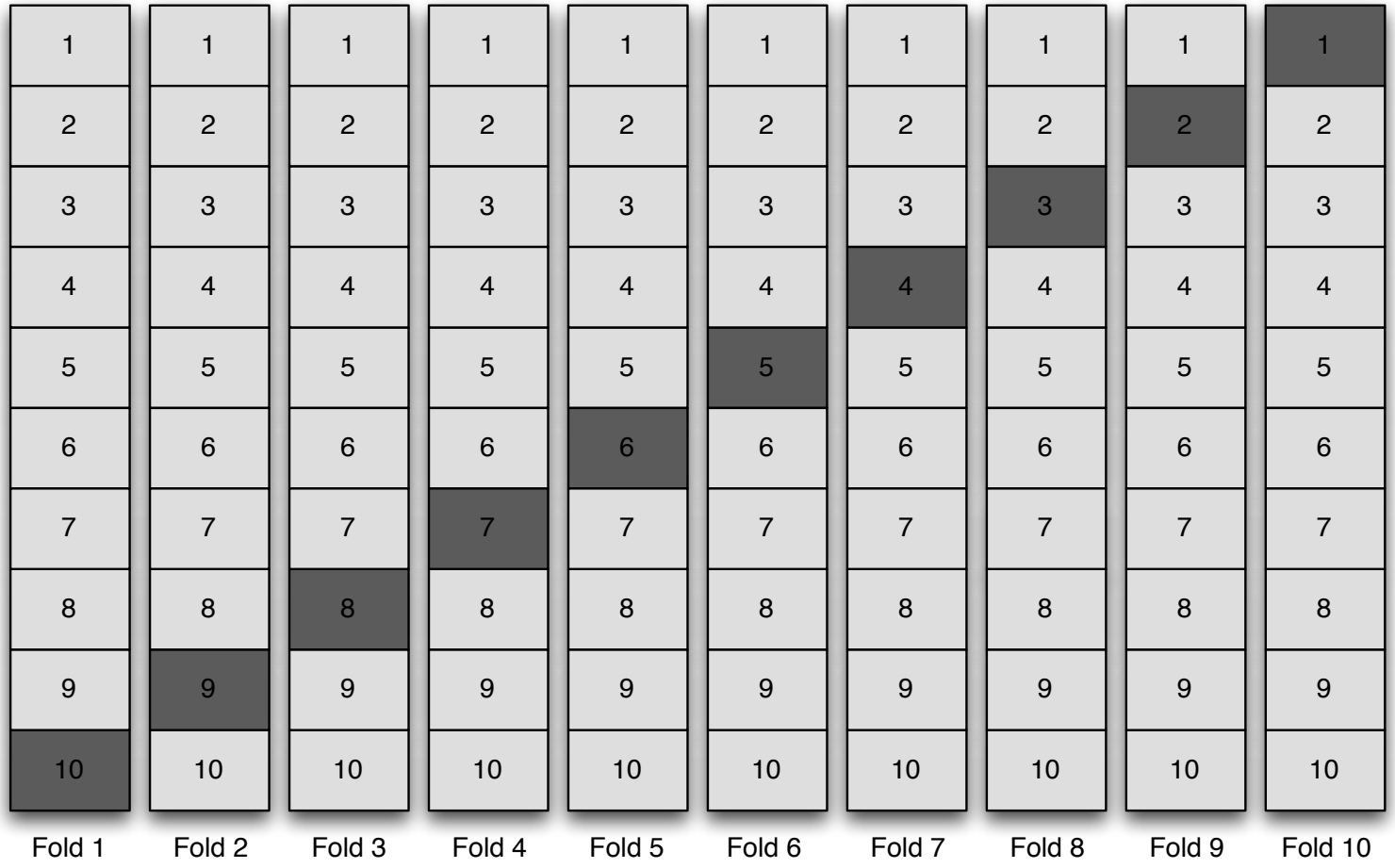
V-fold Cross-Validation

- 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 (in terms of mean squared error) 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.



Expanding the library of candidate algorithms

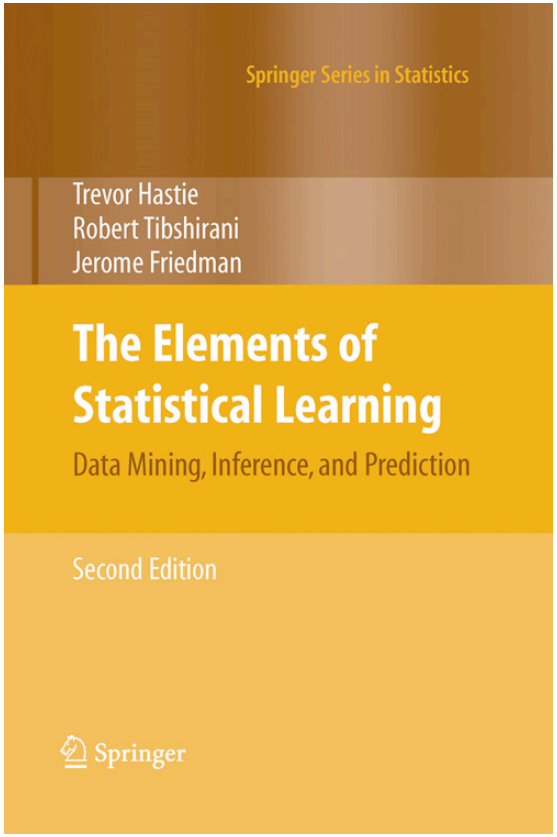
- When selecting candidates, we don't have to limit ourselves to parametric regression models
  - All we need for a given candidate is that it takes as input our observed data and gives as output a prediction function
- Lots of fancier approaches are out there
  - Many of these approaches are themselves 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
- Deletion/Substitution/Addition
- Multiple Additive Regression Splines (MARS)
- Random Forests
- Bagging- Bootstrap aggregation of trees
- Neural networks
- Least Angle Regression (LARS)
- Polynomial spline regression
- …

A good reference



The R Library of Prediction Algorithms

- The key is a good library of machine learning algorithms
- Currently 41 R packages for machine learning/ prediction

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

- Can expand this further by using
  - Different tuning parameters
  - Different candidate covariates/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

- 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 just 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-validiation

- 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
- We will do as well (asymptotically) 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

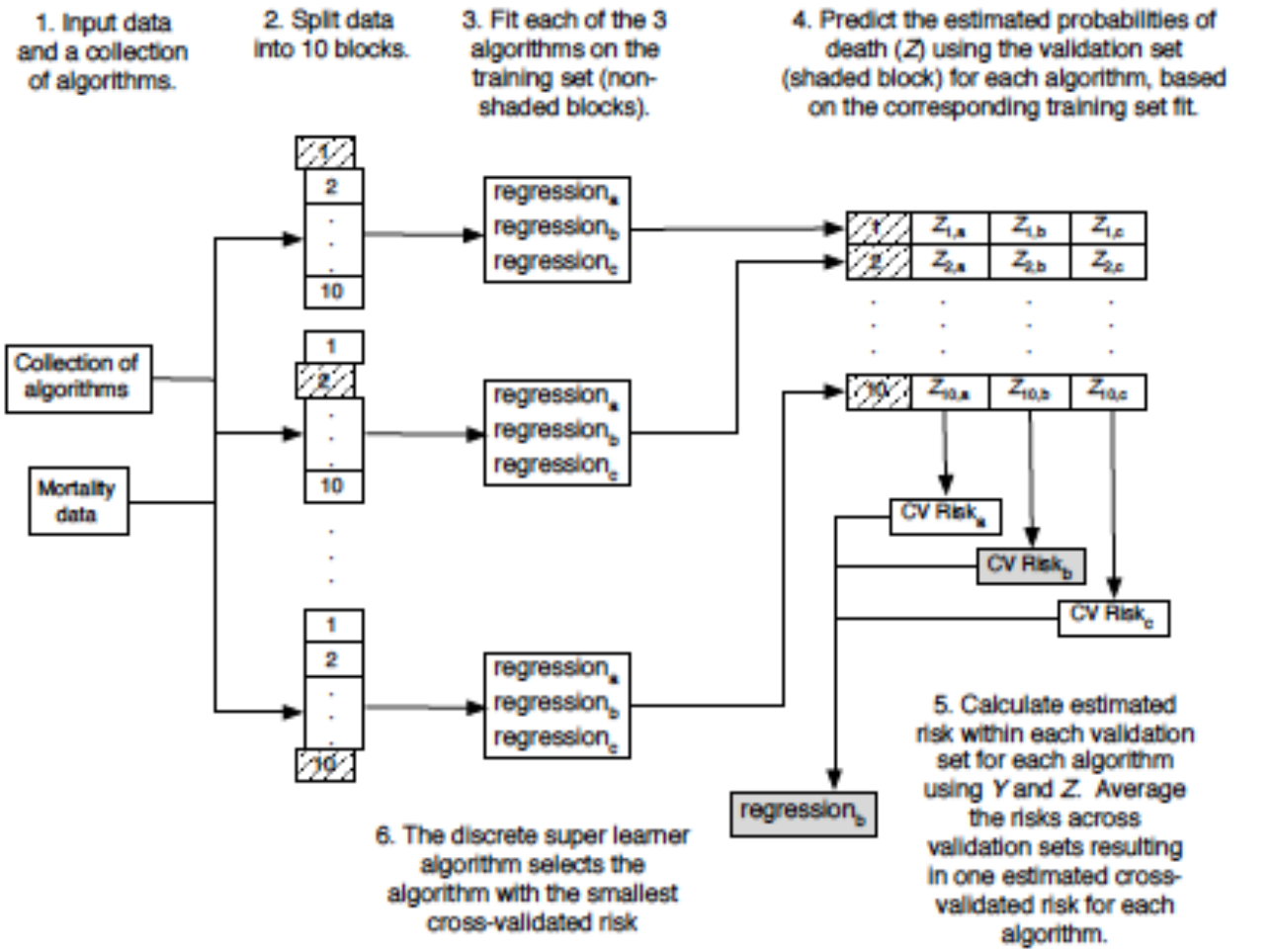
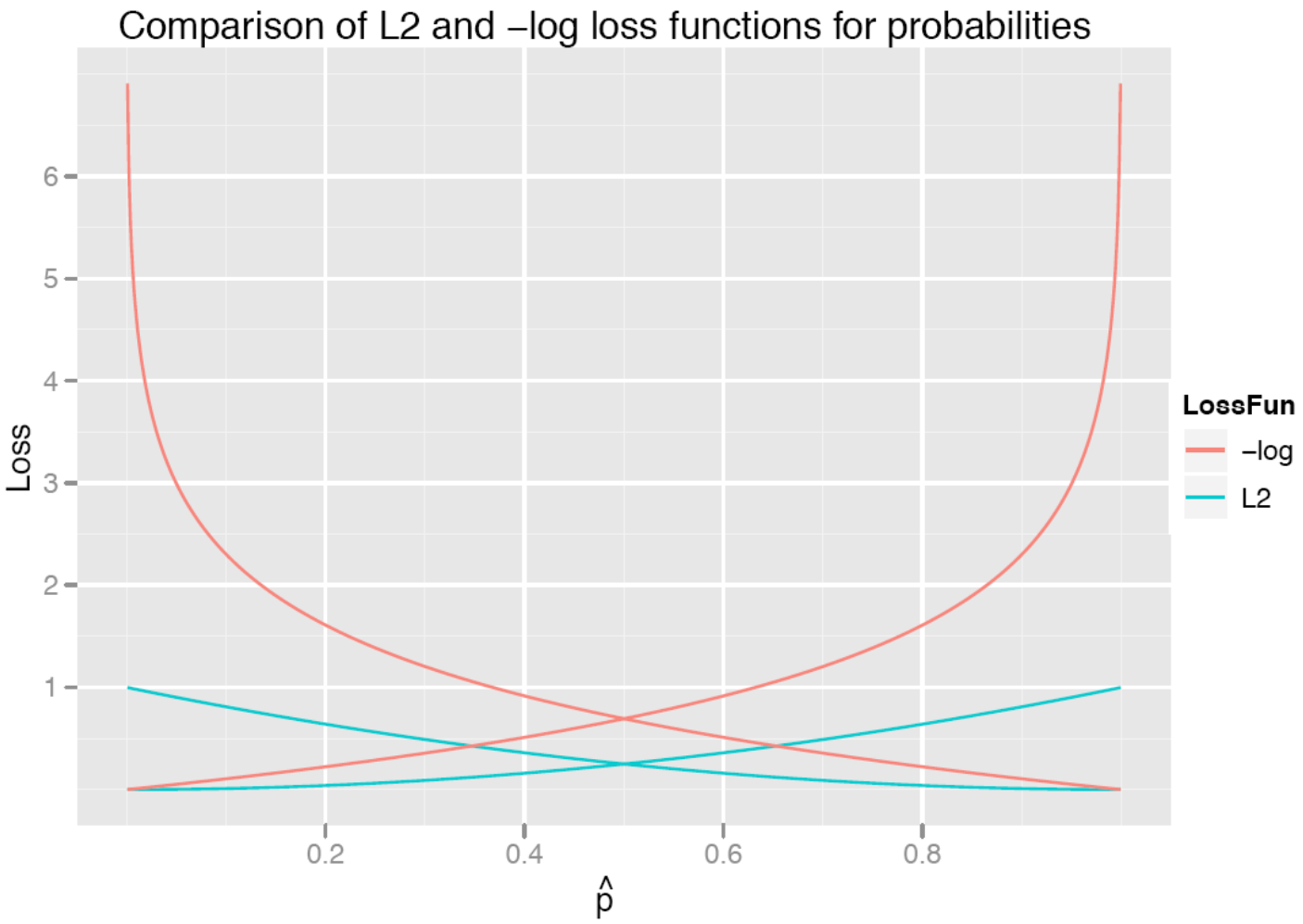


Fig. 3.1 Discrete super learner algorithm for the mortality study example where $Q_n^b(A, W)$ is the algorithm with the smallest cross-validated risk

TLB Chapter 3

Loss Function Must be Bounded



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)

Beating the Best Algorithm

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

Super Learner

- Works on a library of candidates (eg regression fits) created by running each of the competing algorithms

  on the training data
- Rather than just using choosing the best fit, it creates a weighted (convex) combination of the fits

Super Learner

- The weights themselves are fit data-adaptively using cross-validation to give the best overall fit
  - Which weighted combination has the lowest cvRisk, over the family of weighted combinations?
- Can think of this as a way of building an even larger library

Combining the Prediction Models

1. Use each algorithm-specific model (fit in the training sample) to get a predicted outcome for patients in the validation sample
2. Regress the observed outcomes in the validation sample on the (algorithm-specific) predicted outcomes
   - Dependent variable=Observed outcome
   - Independent variables=Algorithms
     - Values=Algorithm-specific predictions
3. The coefficients from this regression are used to weight the contribution of each algorithm to the final prediction model

Combining the Prediction Models

- Intuition: The better the algorithm
  - The closer its predicted outcome is to the observed outcome
  - The larger its coefficient in the regression
  - The larger its weight in the final predictor
- User can specify parametric family of weighted combinations to consider
  - It turns out that simple regression is not stable
  - Typically constrain the regression model to be a convex combination
    - All coefficients are positive
    - Coefficients sum to 1
    - No intercept

Super Learner Flow Chart

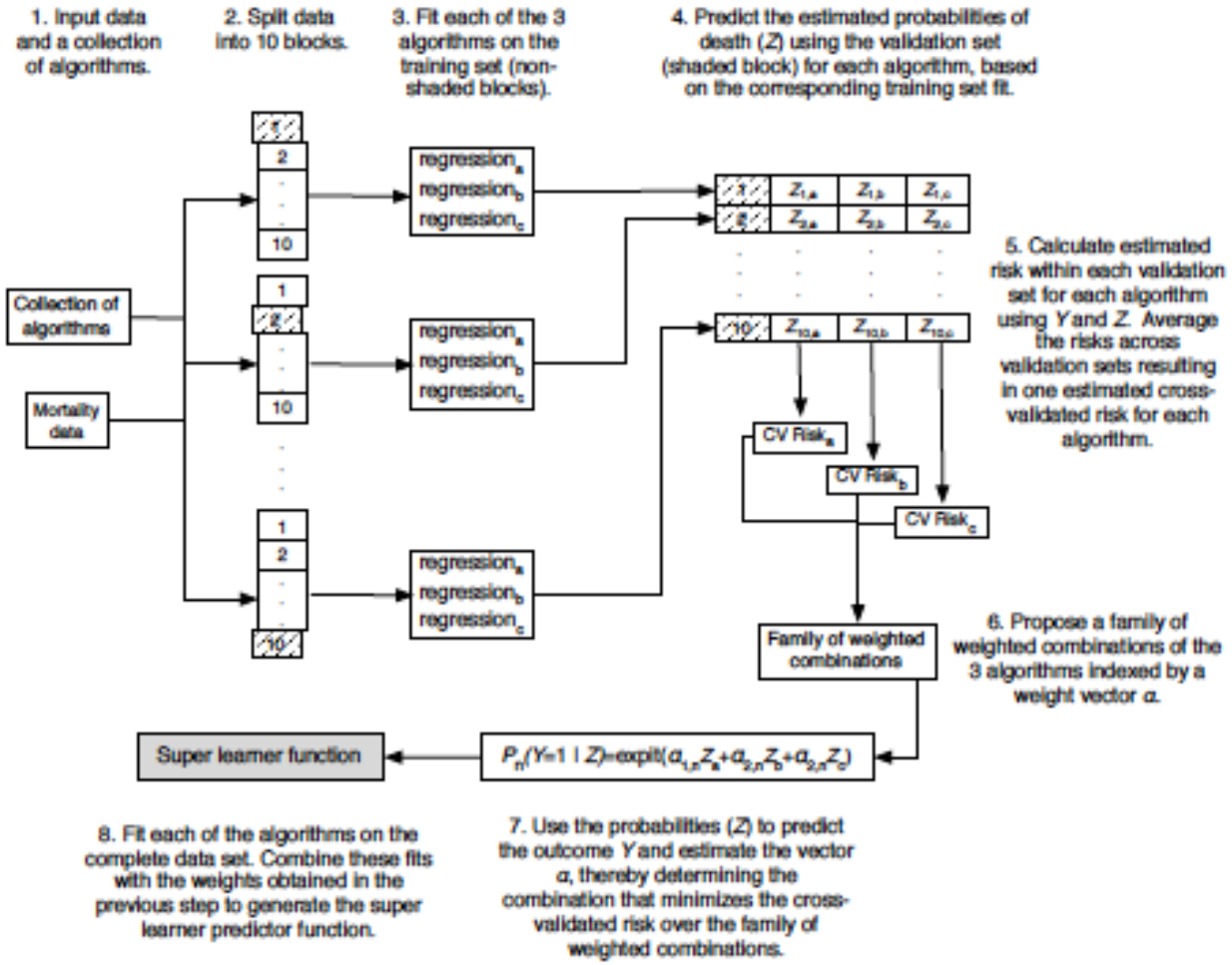


Fig. 3.2 Super learner algorithm for the mortality study example

TLB Chapter 3

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

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]

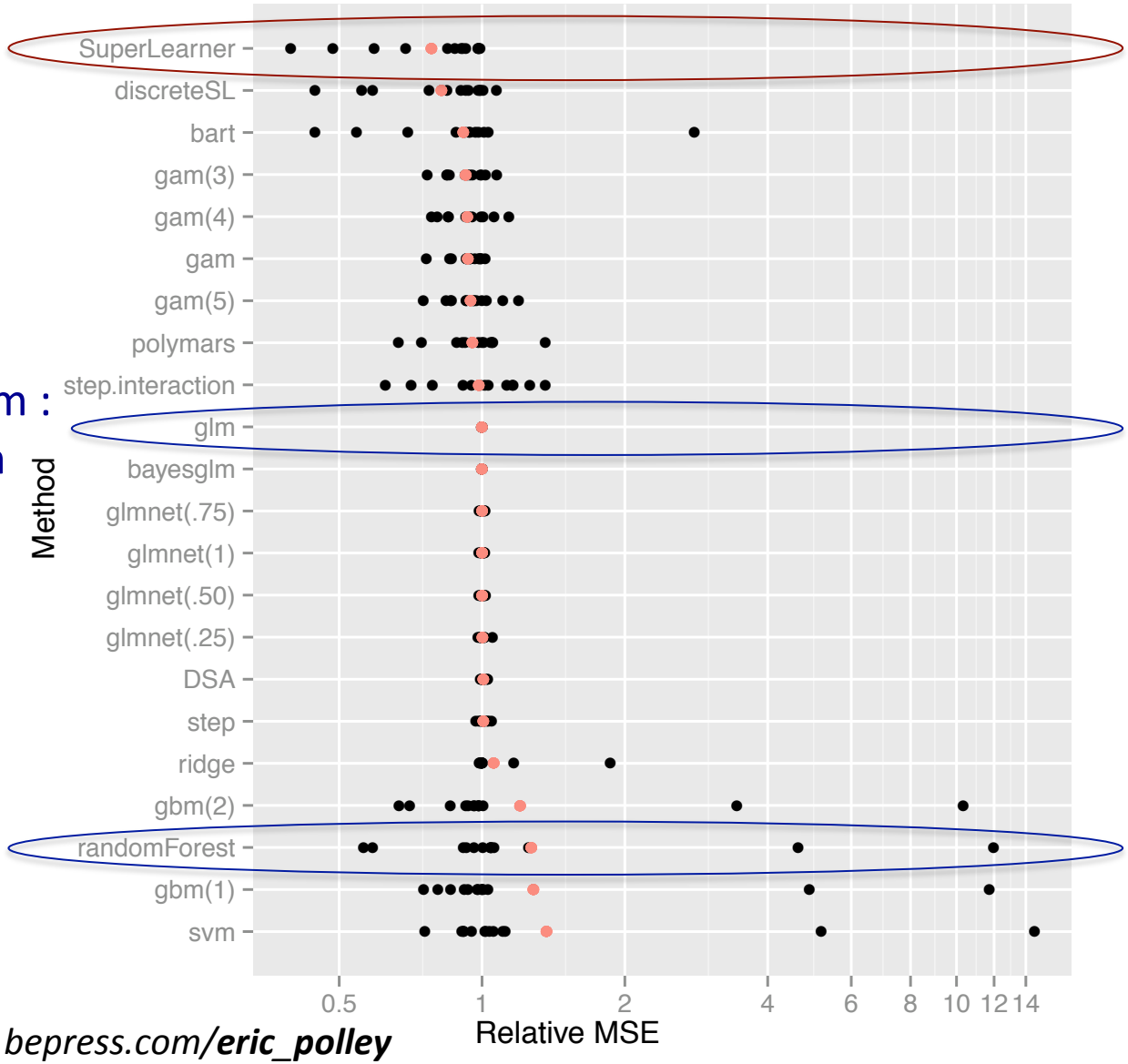
Technical Report: *works.bepress.com/**eric_polley***

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***

Summary: Oracle Results

- Requires that
  - Loss Function is bounded
  - Number of algorithms in the library is polynomial in sample size
- If none of the algorithms converges at a parametric rate
  - Superlearner performs asymptotically as well as the oracle selector (which chooses the best weighted combination of the algorithms)
- If one of algorithms converges at a parametric rate
  - Superlearner still achieves the almost parametric rate of convergence *log n/n*

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
  - 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
   - 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 repressions
- Discrete Super Learner: Choose the algorithm with the lowest cross validated risk
  - Ex. lowest cross validated MSE
- Super Learner: Choose the convex combination with the lowest cross validated risk
- Additional layer of cross validation to evaluate the performance of Super Learner