Modern Statistical Learning Methods for Observational Data and Applications to Comparative Effectiveness Research

Chapter 3: Super Learning

David Benkeser
Emory Univ.

Marco Carone
Univ. of Washington

Larry Kessler
Univ. of Washington

Module 14

4th Annual Summer Institute for Statistics in Clinical Research

07/27/2017

Contents of this chapter

1. Understanding bias/variance tradeoff in estimation
2. Estimation via cross-validated risk assessment
3. Super learning and oracle inequalities

Understanding bias/variance tradeoff in estimation

So far, we discussed ATE estimation via G-computation or IPTW. This hinged on being able to estimate a regression function well.

- Empirical moment approach works well if W is low-dimensional and discrete.
- If W has continuous components, the estimator breaks down.

Generally, we must borrow information across W levels to learn about these functions.

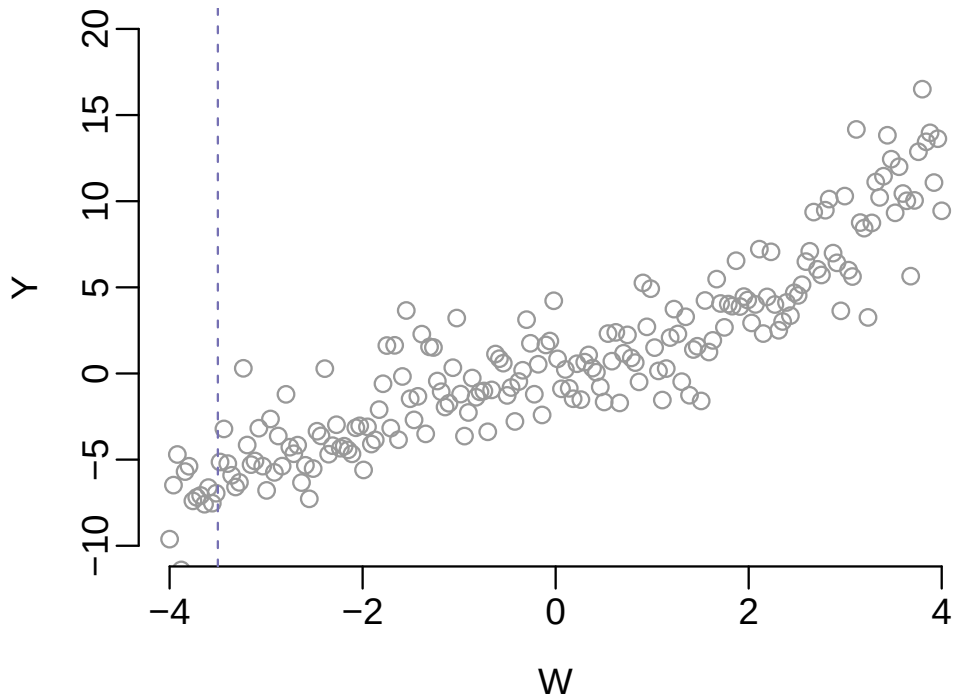
- Key question: how much borrowing is just the right amount?

If we borrow little information across participants, then our estimator is only using a small portion of the data:

- info borrowed from very similar patients $\Rightarrow$ low bias;
- fewer patients to borrow info from $\Rightarrow$ high variance.

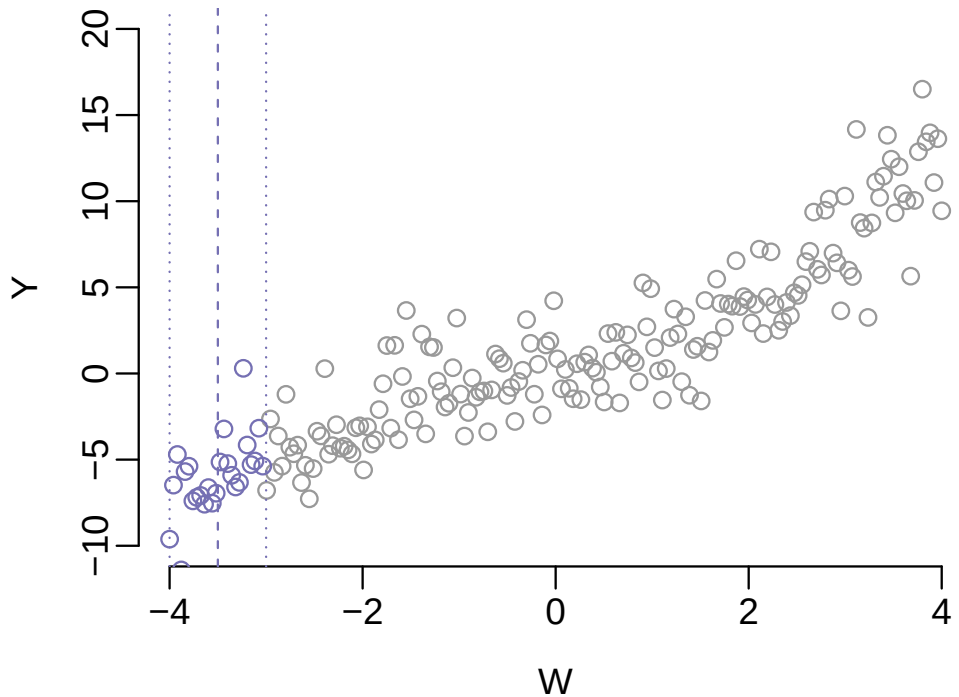
Understanding bias/variance tradeoff in estimation

Consider estimating $E(Y \mid A = 1, W = -3.5)$.



Understanding bias/variance tradeoff in estimation

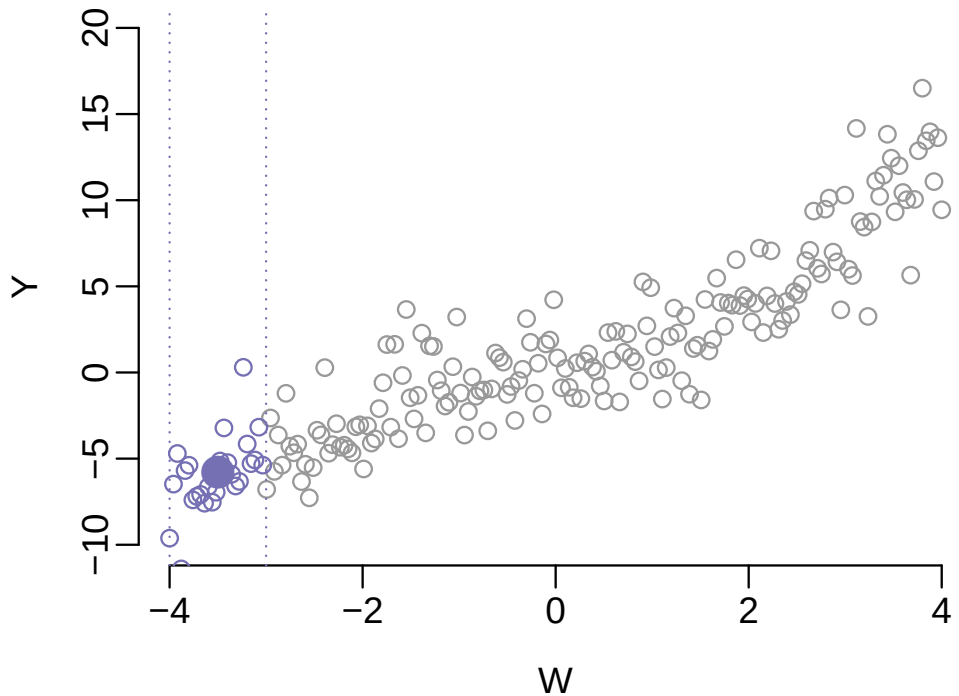
We could use patients with $-4 \leqslant W \leqslant -3$ to learn about those with $W = -3.5$.



Understanding bias/variance tradeoff in estimation

This motivates the uniform kernel estimator with h-sized window:

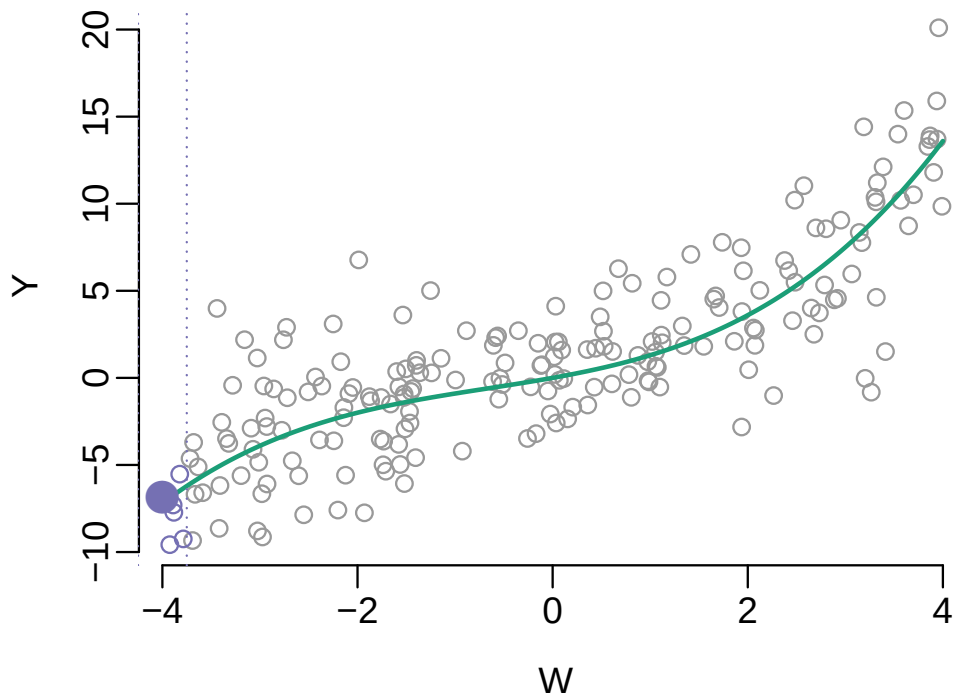
$$\widehat{E}(Y \mid A = 1, W = w) \;=\; \text{average outcome among patients with } |W - w| \leqslant \frac{h}{2}$$



Understanding bias/variance tradeoff in estimation

How big should h be?

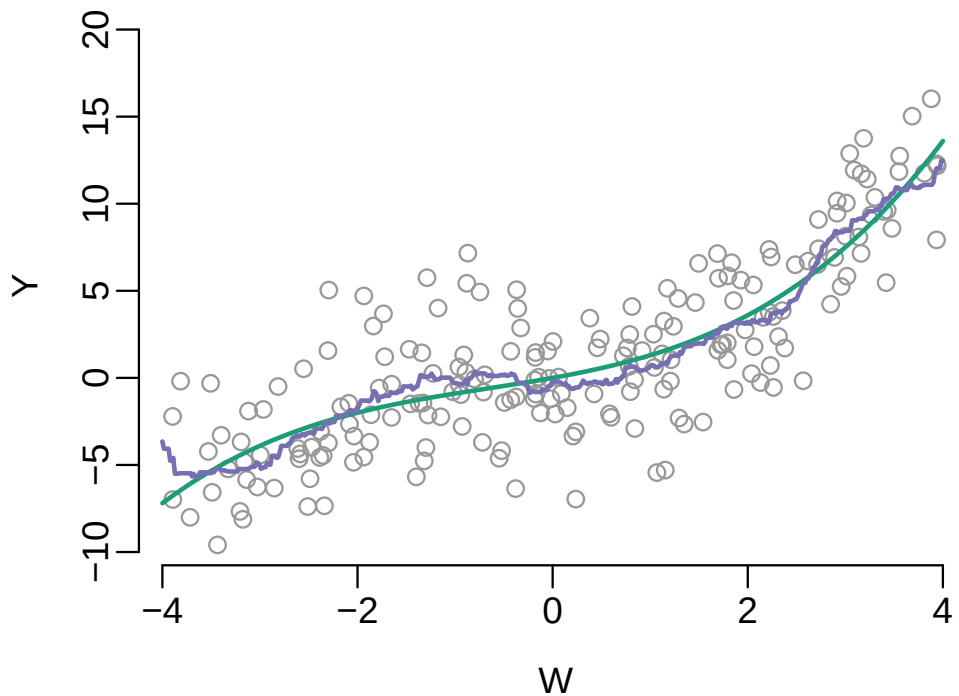
- Small $h \Rightarrow$ borrowing info from very similar patients $\Rightarrow$ low bias.



Understanding bias/variance tradeoff in estimation

How big should h be?

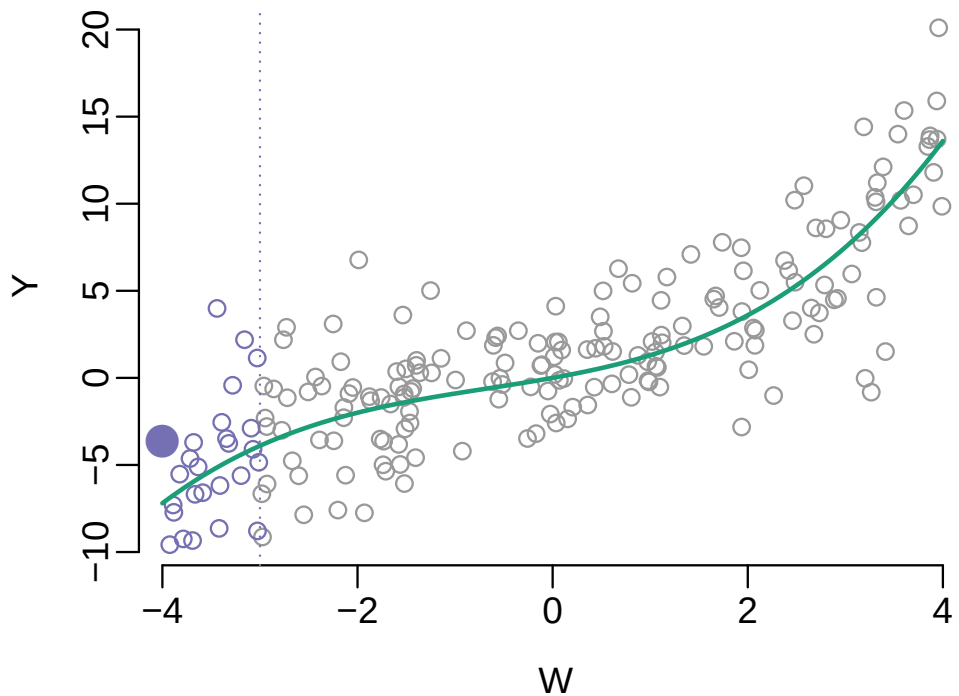
- Small $h \Rightarrow$ fewer patients to borrow info from $\Rightarrow$ high variance.



Understanding bias/variance tradeoff in estimation

How big should h be?

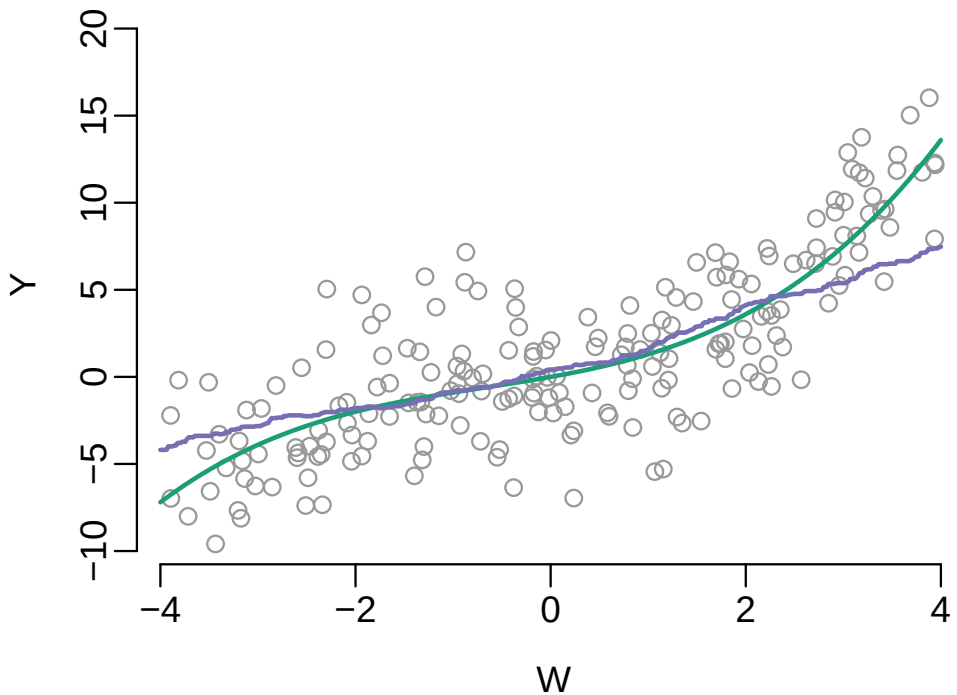
- Big $h \Rightarrow$ borrowing info from less similar patients $\Rightarrow$ high bias.



Understanding bias/variance tradeoff in estimation

How big should h be?

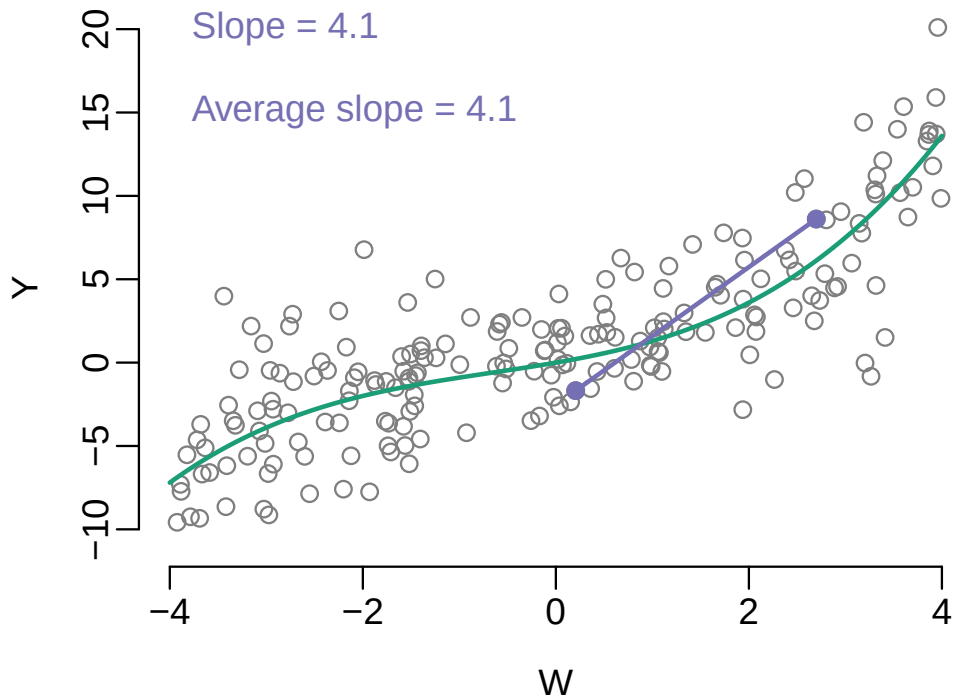
- Big $h \Rightarrow$ more patients to borrow info from $\Rightarrow$ low variance.



Understanding bias/variance tradeoff in estimation

Linear regression is an example of a method that borrows heavily.

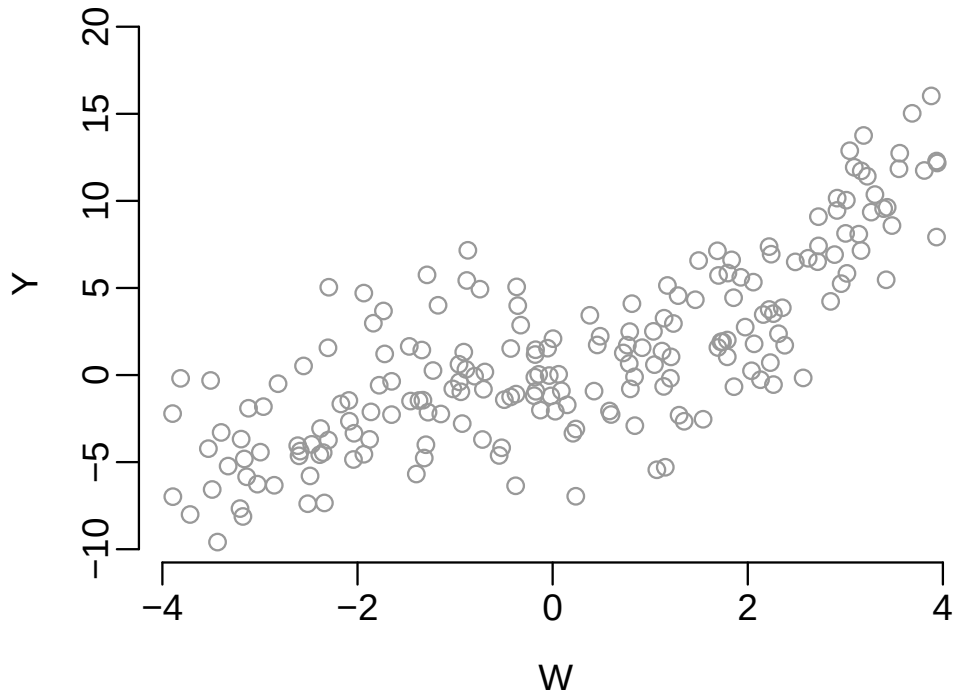
- It turns out $\hat{\beta}$ is a weighted average of all pairwise slopes.
- Weights equal to squared distance between observations.



Understanding bias/variance tradeoff in estimation

Linear regression is an example of a method that borrows heavily.

- low variance, but potentially high bias.



Understanding bias/variance tradeoff in estimation

We have many different regression tools that tradeoff bias and variance to different degrees and in different ways.

- (nonparametric) empirical moment, kernel regression, neural networks, random forests;
- (semiparametric) generalized additive models, partially linear additive models;
- (parametric) linear regression, logistic regression, spline regression.

For any single regression method, the tradeoff between bias and variance is generally governed by modeling choices and/or tuning parameters.

- linear regression with linear versus 4^{th} degree polynomial terms;
- uniform kernel estimator with large versus small window h;
- regression tree with maximum depth one versus thirty.

The best bias/variance tradeoff depends on the (unknown) true regression function.

Estimation via cross-validated risk assessment

How can we use the observed data to learn about the true relationship?

Why not try out a bunch of different regressions and see which one fits best?

- This is what we will do, but we must be careful!
- Estimators must generally be pre-specified.
- What do we mean by "fits best"?
- How do we evaluate this fit appropriately?

We need a careful framework for candidate regression evaluation to avoid bias.

- Pick the method that gives an answer that makes sense (e.g., has small p-value)?
- If sample size is huge, we might find an effect even if there is no effect.

We also risk producing misleading inference about our estimate of the effect.

- Standard confidence intervals and p-values require complete pre-specification.
- They will usually not account for uncertainty in trying many models.

Estimation via cross-validated risk assessment

As an example, the G-computation ATE estimator can be implemented in two steps:

1. Use specified regression technique to estimate outcome regression $E(Y \mid A, W)$.
2. Plug resulting estimate into G-computation formula.

What if we replace 1 with:

1a. Evaluate a collection of different pre-specified regression techniques;
1b. Choose the "best" technique as final regression estimate $\bar{Q}_n$?

We can still entirely pre-specify the estimation procedure.

- We need to determine how to pick the "best" technique.

Can we have a fair competition between estimators to determine the best?

Estimation via cross-validated risk assessment

How will we score our estimators to determine which is closest to the truth?

We focus on the outcome regression; same arguments for the propensity score.

A loss function L measures how far the prediction of Y made by $\bar{Q}_n$ is from the actual value of Y. Common loss functions used include:

$$\text{squared error loss:} \quad L(\bar{Q}_n)(w, a, y) := \{y - \bar{Q}_n(a, w)\}^2;$$

$$\text{log-likelihood loss:} \quad L(\bar{Q}_n)(w, a, y) := -\log\left[\bar{Q}_n(a, w)^y (1 - \bar{Q}_n(a, w)\}^{1-y}\right]$$

The average loss of a candidate estimator $\bar{Q}_n$ is called its risk:

$$\begin{aligned} R(\bar{Q}_n) &:= E[L(\bar{Q}_n)(W, A, Y)] \\ &= \sum_{w,a,y} L(\bar{Q}_n)(w, a, y) P(W = w, A = a, Y = y) \ . \end{aligned}$$

The risk can used to score the performance of different candidate estimators.

Estimation via cross-validated risk assessment

How do we know that risk is a good way to keep score?

We say that L is a valid loss function for estimating the true outcome regression $\bar{Q}$ if choosing the estimator $\bar{Q}_n = \bar{Q}$ results in the minimum possible risk.

Example: minimizing mean squared error

$$\begin{aligned} R(\bar{Q}_n) &= E\left[Y - \bar{Q}_n(A,W)\right]^2 \\ &= E\left[Y - \bar{Q}(A,W) + \bar{Q}(A,W) - \bar{Q}_n(A,W)\right]^2 \\ &= E\left[Y - \bar{Q}(A,W)\right]^2 + 2E\left\{\left[Y - \bar{Q}(A,W)\right]\left[\bar{Q}(A,W) - \bar{Q}_n(A,W)\right]\right\} \\ &\qquad + E\left[\bar{Q}(A,W) - \bar{Q}_n(A,W)\right]^2 \\ &= \underbrace{E\left[Y - \bar{Q}(A,W)\right]^2}_{\text{always} \geqslant 0} + \underbrace{E\left[\bar{Q}(A,W) - \bar{Q}_n(A,W)\right]^2}_{\text{when is this 0?}} \end{aligned}$$

An estimator with a small risk is preferred – just like in golf, we prefer lower scores!

Unsurprisingly, the estimator with the smallest risk is the true outcome regression.

Estimation via cross-validated risk assessment

How does this relate to the bias/variance tradeoff?

$$R(\bar{Q}_n) = E\{[Y - \bar{Q}(A, W)]^2\} + E\{[\bar{Q}(A, W) - \bar{Q}_n(A, W)]^2\}$$

The second term can be written as

$$\begin{aligned} & E\left[\bar{Q}(A, W) - \bar{Q}_n(A, W)\right]^2 \\ & \quad = E\left\{\bar{Q}(A, W) - E[\bar{Q}_n(A, W)] + E[\bar{Q}_n(A, W)] - \bar{Q}_n(A, W)\right\}^2 \\ & \quad = E\left\{\bar{Q}(A, W) - E[\bar{Q}_n(A, W)]\right\}^2 + E\left\{E[\bar{Q}_n(A, W)] - \bar{Q}_n(A, W)\right\}^2 \end{aligned}$$

Combining these, we see that the risk $R(\bar{Q}_n)$ can be decomposed as

$$\underbrace{E\left[Y - \bar{Q}(A, W)\right]^2}_{\text{noise}} + \underbrace{E\left\{\bar{Q}(A, W) - E[\bar{Q}_n(A, W)]\right\}^2}_{\text{bias squared}} + \underbrace{E\left\{E[\bar{Q}_n(A, W)] - \bar{Q}_n(A, W)\right\}^2}_{\text{variance}}.$$

Mean squared error is the sum of noise inherent to the data, bias of the estimator, and variance of the estimator.

Estimation via cross-validated risk assessment

How can we estimate the risk of an estimator?

Idea: Use empirical risk estimate $R_n(\bar{Q}_n) = \frac{1}{n} \sum_{i=1}^{n} \left\{ Y_i - \bar{Q}_n(A_i, W_i) \right\}^2$.

Problem: $\bar{Q}_n$ was fit using the same data that we use to estimate its risk!

Analogy: a student got a glimpse of the exam before taking it.

- focused on learning test questions very well;
- test result does not reflect how well student has learned the subject.

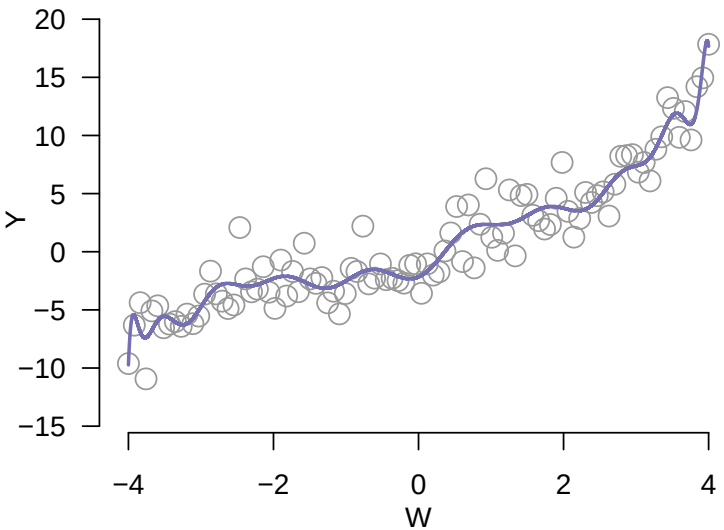
Consider fitting to $n = 100$ observations the linear model $W + W^2 + \cdots + W^{20}$

$$\bar{Q}_n(1, w) = \beta_{0,n} + \beta_{1,n} w + \beta_{2,n} w^2 + \cdots + \beta_{20,n} w^{20}$$

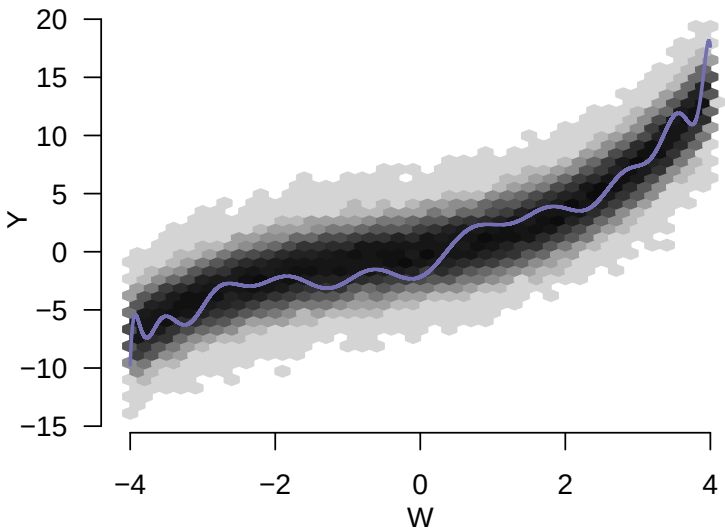
versus fitting the true linear model $W + W^2 + W^3$

$$\bar{Q}_n(1, w) = \beta_{0,n} + \beta_{1,n} w + \beta_{2,n} w^2 + \beta_{3,n} w^3.$$

Estimation via cross-validated risk assessment

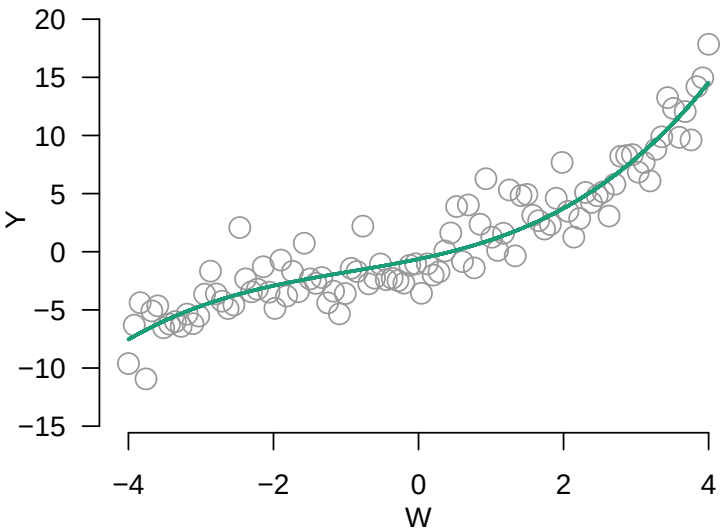


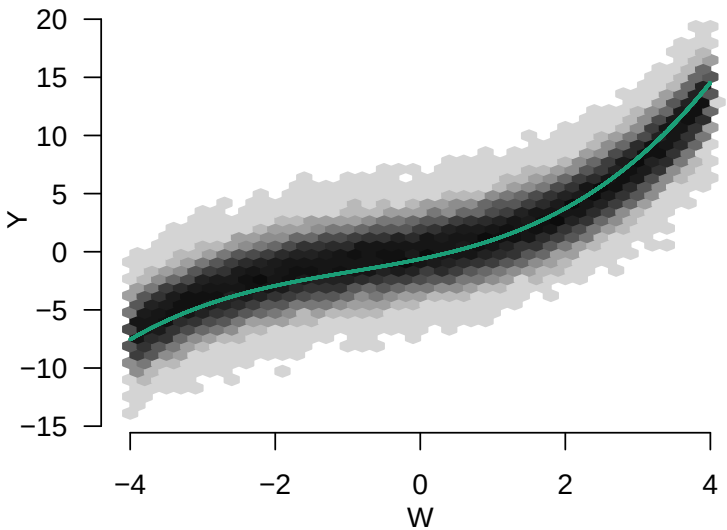
$R_n(\bar{Q}_n) = 2.7$



$R(\bar{Q}_n) = 5.3$

Estimation via cross-validated risk assessment





$R_n(\bar{Q}_n) = 3.7$

$R(\bar{Q}_n) = 4.4$

Estimation via cross-validated risk assessment

This illustrates two key points:

1. we are overly optimistic about how well we are fitting each estimator;
2. we would select the wrong estimator.

What we really need is external data to properly evaluate estimators.

- We could get a better idea about how well our estimators are doing.
- We could objectively select the correct estimator to use.
- . . . but such data are rarely available.

Instead, we use cross-validation to estimate risk.

- Fit regression on part of the data, evaluate on another part.
- Mimics evaluating fit on external data.
- Many forms of cross-validation; we focus on V-fold cross-validation.

Estimation via cross-validated risk assessment

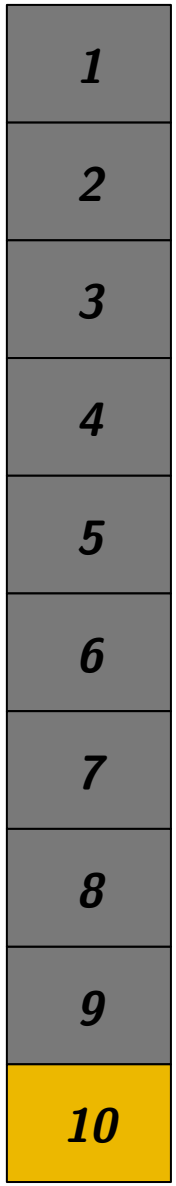
Data are divided into V sets of size $\sim \frac{n}{V}$. Here, $V = 10$.

Fold 1 = training sample $\mathcal{T}_1$ + validation sample $\mathcal{V}_1$.

- Training sample is used to fit ("train") the regressions.
- Validation sample is used to estimate risk ("validate").

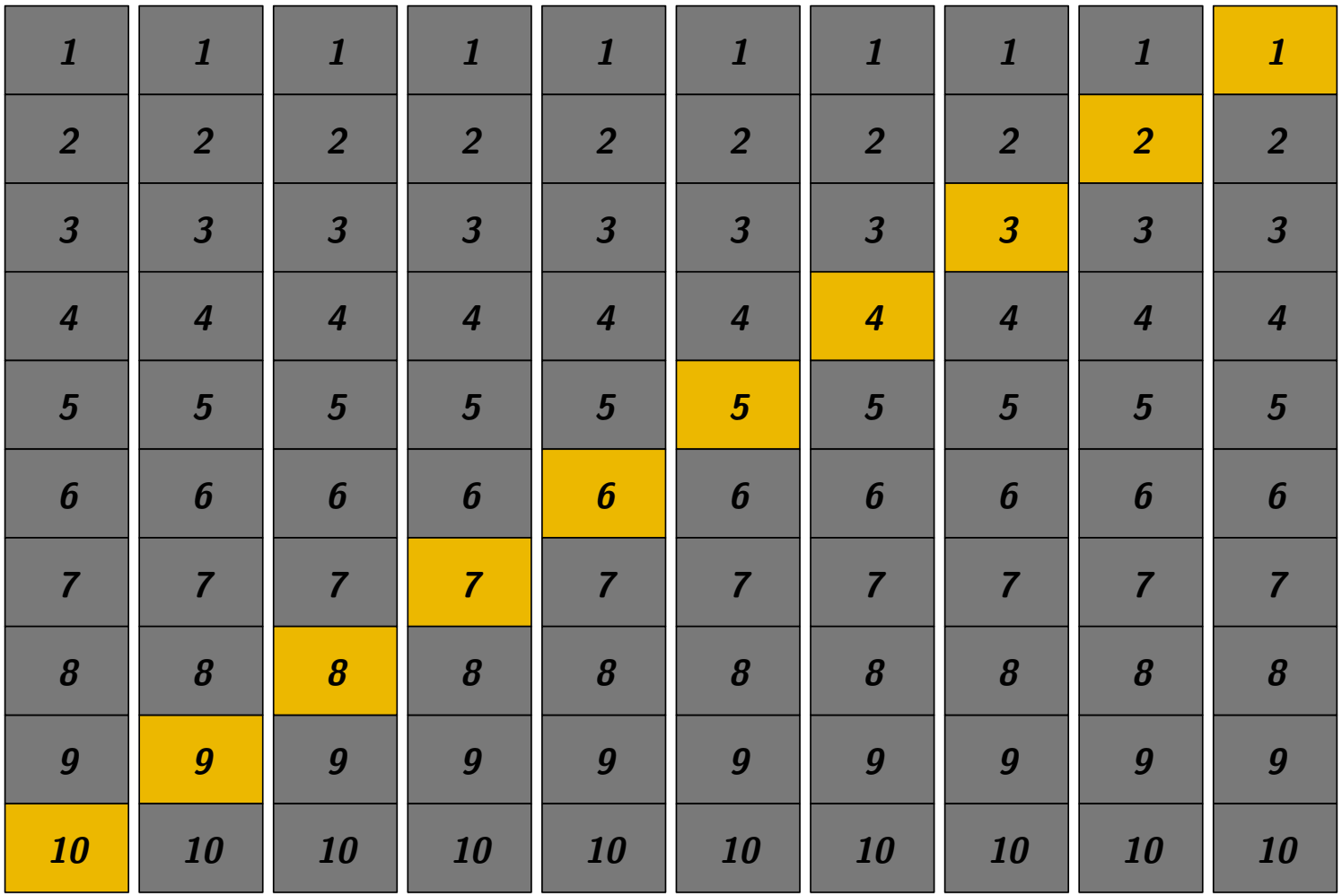
Several factors to consider when choosing V:

- large V = more data to fit regressions (helpful in small datasets or with high-dimensional covariates);
- small V = more data to evaluate risk (helpful for outcomes with skewed distributions);
- large V = greater computation time.



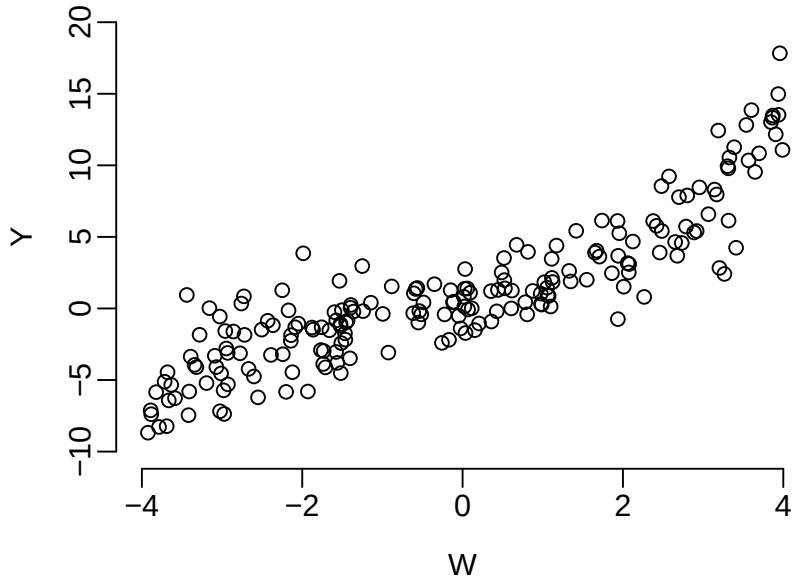
Estimation via cross-validated risk assessment

The validation set rotates until each set has been used as validation once.



Estimation via cross-validated risk assessment

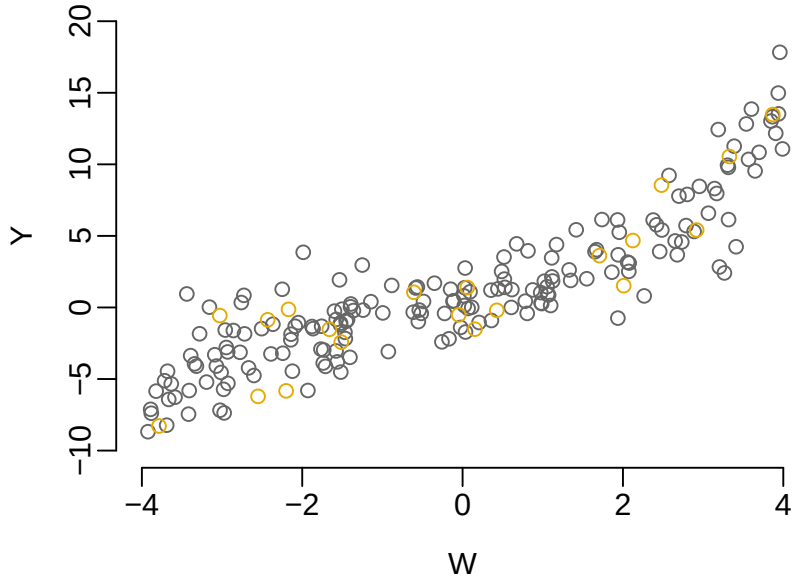
Consider cross-validation for our problem.



Estimation via cross-validated risk assessment

Consider cross-validation for our problem.

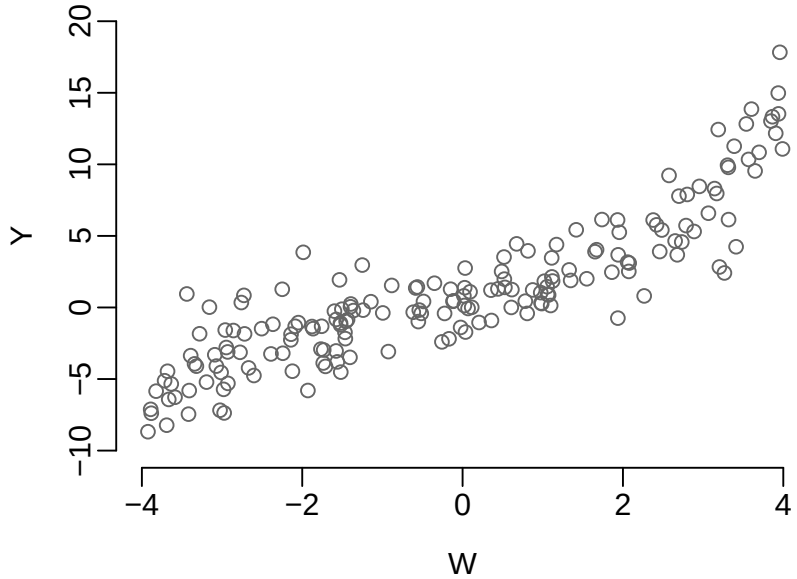
1. Define first training sample $\mathcal{T}_1$ and validation sample $\mathcal{V}_1$.

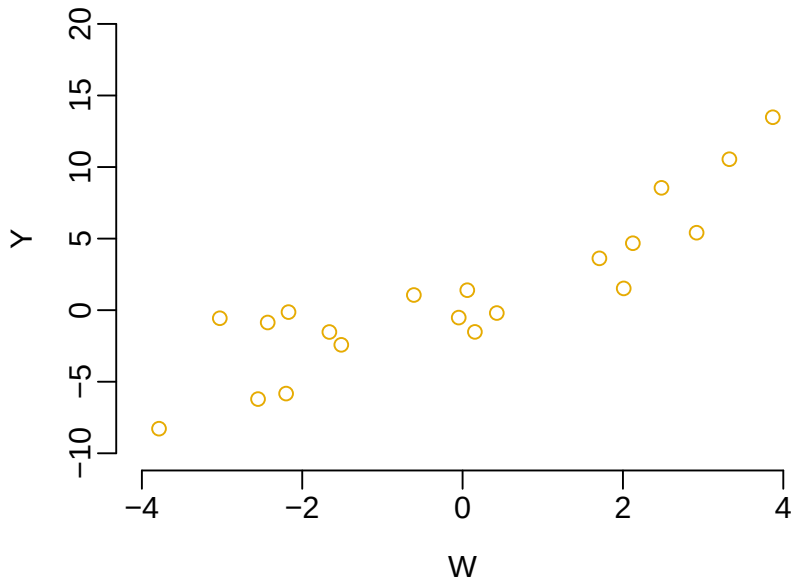


Estimation via cross-validated risk assessment

Consider cross-validation for our problem.

1. Define first training sample $\mathcal{T}_1$ and validation sample $\mathcal{V}_1$.
2. Split data into training sample $\mathcal{T}_1$ (size n_t) and validation sample $\mathcal{V}_1$ (size n_v).

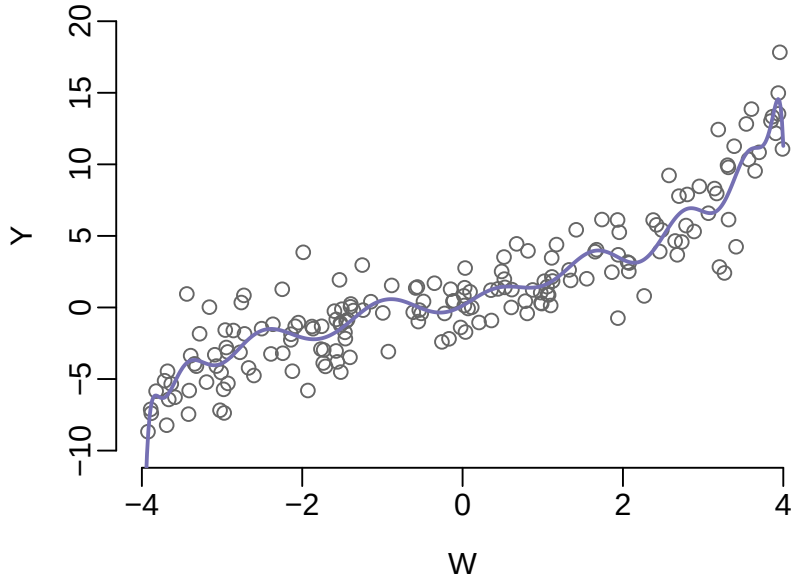


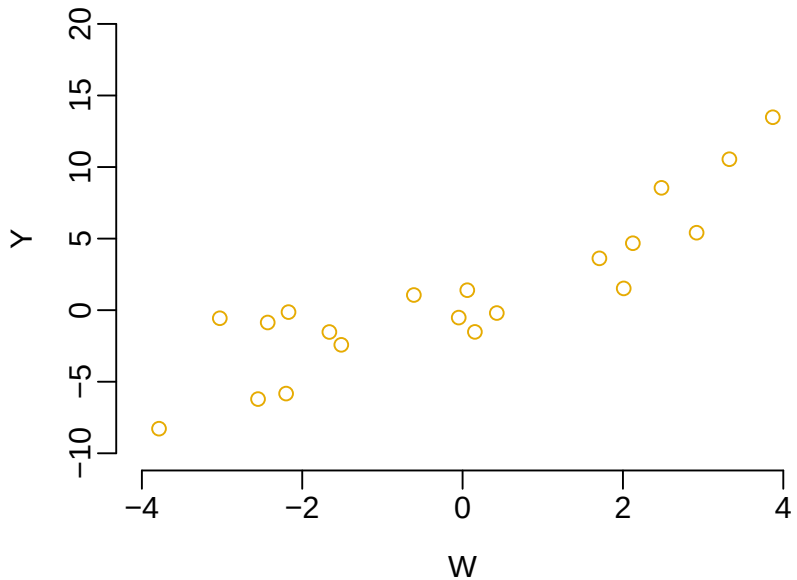


Estimation via cross-validated risk assessment

Consider cross-validation for our problem.

1. Define first training sample $\mathcal{T}_1$ and validation sample $\mathcal{V}_1$.
2. Split data into training sample $\mathcal{T}_1$ (size n_t) and validation sample $\mathcal{V}_1$ (size n_v).
3. Fit $\bar{Q}_{n,1}$ in training sample $\mathcal{T}_1$.



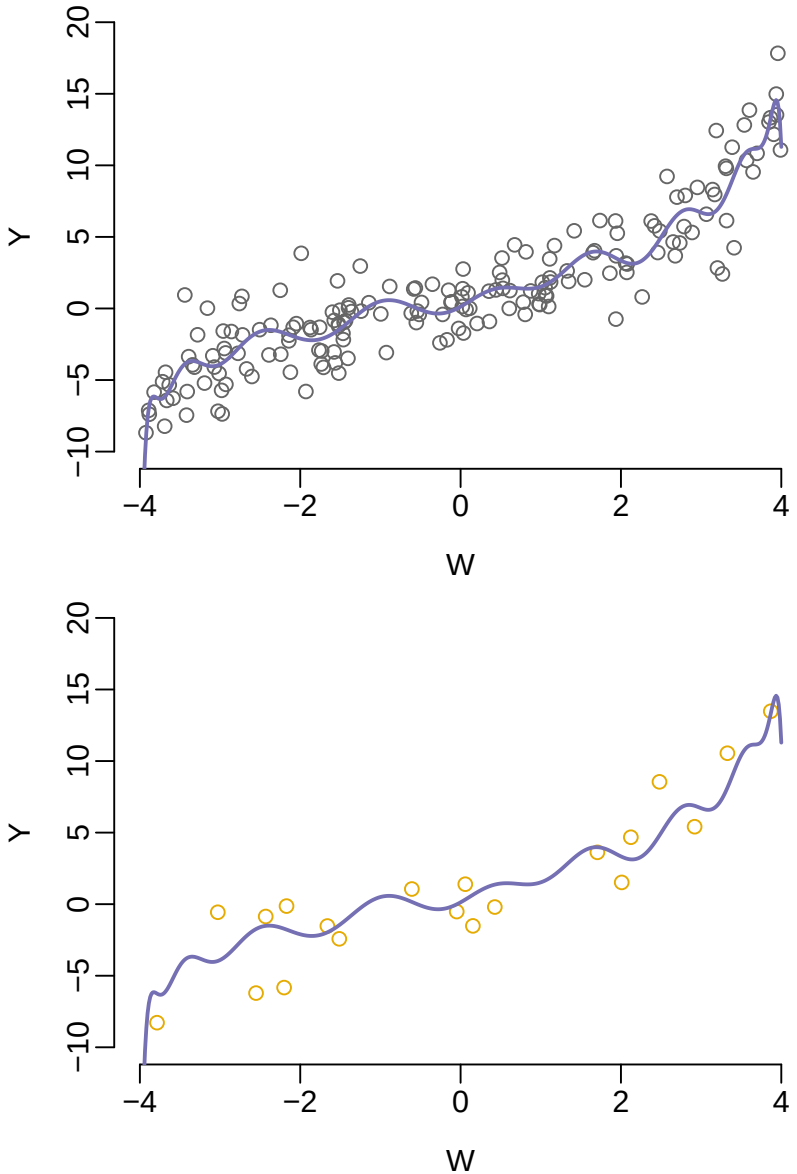


Estimation via cross-validated risk assessment

Consider cross-validation for our problem.

1. Define first training sample $\mathcal{T}_1$ and validation sample $\mathcal{V}_1$.
2. Split data into training sample $\mathcal{T}_1$ (size n_t) and validation sample $\mathcal{V}_1$ (size n_v).
3. Fit $\bar{Q}_{n,1}$ in training sample $\mathcal{T}_1$.
4. Estimate risk in validation sample $\mathcal{V}_1$

$$R_{n,1}(\bar{Q}_{n,1}) = \frac{1}{n_v} \sum_{i \in \mathcal{V}_1} \left\{ Y_i - \bar{Q}_{n,1}(1, W_i) \right\}^2 .$$

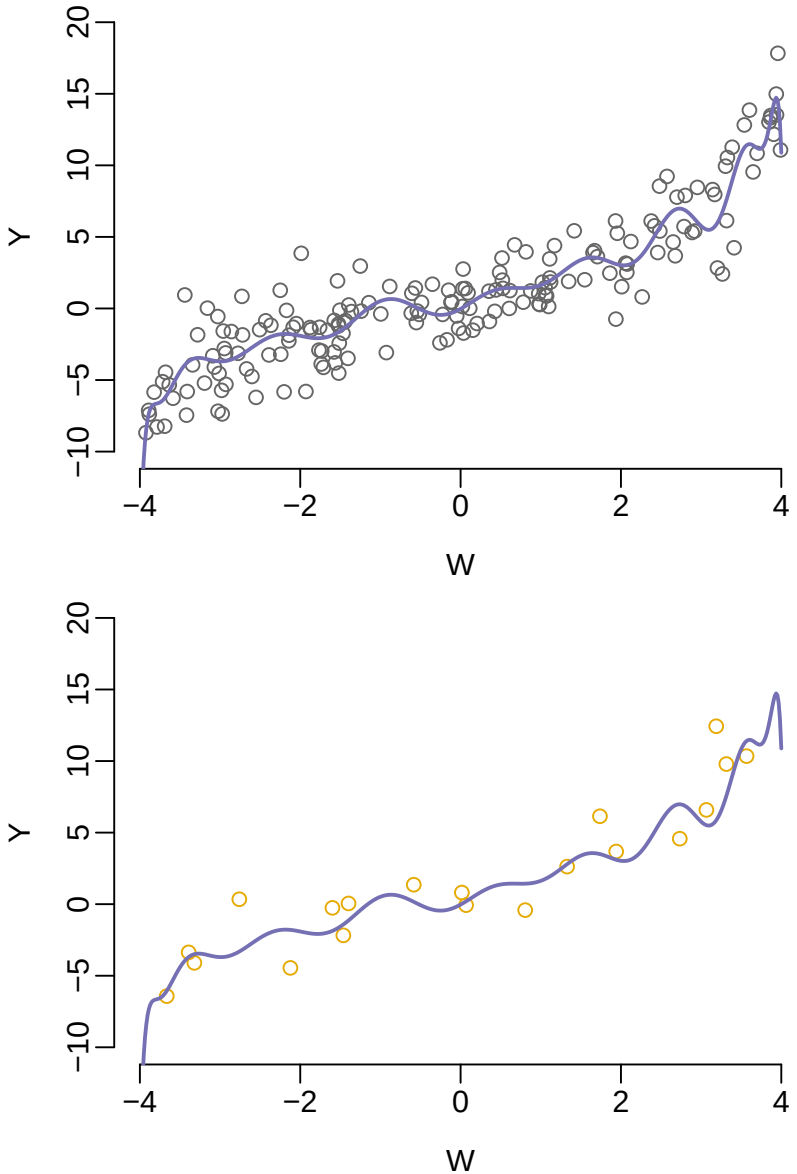


Estimation via cross-validated risk assessment

Consider cross-validation for our problem.

1. Define second training sample $\mathcal{T}_2$ and validation sample $\mathcal{V}_2$.
2. Split data into training sample $\mathcal{T}_2$ (size n_t) and validation sample $\mathcal{V}_2$ (size n_v).
3. Fit $\bar{Q}_{n,2}$ in training sample $\mathcal{T}_2$.
4. Estimate risk in validation sample $\mathcal{V}_2$

$$R_{n,2}(\bar{Q}_{n,2}) = \frac{1}{n_v} \sum_{i \in \mathcal{V}_2} \left\{ Y_i - \bar{Q}_{n,2}(1, W_i) \right\}^2 .$$



Estimation via cross-validated risk assessment

How can we estimate the risk of an estimator?

The cross-validated risk is the average risk over the folds, which we estimate by

$$R_{CV,n}(\bar{Q}_n) := \frac{1}{V} \sum_{v=1}^{V} R_{n,v}(\bar{Q}_{n,v}) .$$

The estimated cross-validated risk should provide a close approximation of the true risk of an estimator.

We should choose the estimator with the lowest estimated cross-validated risk!

- Idea originated with "model stacking" of Wolpert (1992) and Breiman (1996);
- Rebranded as the discrete super learner due to optimality results (van der Laan, Polley & Hubbard, 2007).

Super learning and oracle inequalities

Say we want to compare different outcome regression estimators $\bar{Q}_{n,k}$, $k = 1, 2, \ldots, K$.

Denote by $\bar{Q}_{n,k_n^{\text{cv}}}$ the estimator with the smallest cross-validated risk.

The best amongst all candidate estimators is the oracle selector

$$k_n^{\text{or}} := \underset{k}{\text{argmin}} \frac{1}{V} \sum_{v=1}^{V} R(\bar{Q}_{n,k,v}) \ .$$

- The oracle estimator depends on the true risk and thus is unknown in practice.
- Serves as a benchmark – the best we could do given these K estimators.

We can compare the performance of $\bar{Q}_{n,k_n^{\text{cv}}}$ to $\bar{Q}_{n,k_n^{\text{or}}}$.

- How well do we mortals do compared to the oracle?
- Compare estimators in terms of difference in risk $d_0(\bar{Q}_n, \bar{Q}) := R(\bar{Q}_n) - R(\bar{Q})$.

Super learning and oracle inequalities

Under conditions, van der Vaart, Dudoit & van der Laan (2006) showed that, for any $\lambda > 0$ and for p denoting proportion in validation sample,

$$\underbrace{\frac{1}{V}\sum_{v=1}^{V} d_0(\bar{Q}_{v,n,k_n^{\text{cv}}}, \bar{Q})}_{\text{super learner vs. truth}} \leqslant (1+2\lambda) \underbrace{\sum_{v=1}^{V} d_0(\bar{Q}_{v,n,k_n^{\text{or}}}, \bar{Q})}_{\text{oracle vs. truth}} + \underbrace{2C(\lambda)\left(\frac{1+\log K}{np}\right)}_{\text{goes to 0 as n grows}}.$$

Some observations on oracle inequality:

1. Discrete super learner is *essentially* as close to the truth as oracle.
2. The number of algorithms K is allowed to be very large.

Many studies have demonstrated excellent performance in practice as well (Rose, 2013; Pirracchio et al., 2015).

Super learning and oracle inequalities

Practically, the oracle inequality suggests that we should include as many different pre-specified estimators as possible.

- (nonparametric) empirical moment, kernel regression, neural networks, random forests;
- (semiparametric) generalized additive models, partially linear additive models;
- (parametric) linear regression, logistic regression, spline regression.

Any given method could result in multiple different potential estimators.

- random forest with different tuning parameters;
- generalized additive models with different knots and degrees;
- linear regression with interactions and stepwise selection;

Certain estimators will work well on some data sets and poorly on others.

- The oracle is the best estimator **for this data set**.
- The super learner performs **as well as** the oracle.

Super learning and oracle inequalities

What about model checking?

When a single parametric regression is used, post-hoc model checking is common.

- Examine residual plots, add/remove terms, refit.
- Inference is based on the final-selected model.

Estimators must generally be pre-specified.

- Any post-hoc checking we might do should also be pre-specified.
- Standard confidence intervals and p-values require complete pre-specification.
- They will usually not account for uncertainty in post-hoc checking.

Super learning avoids the need for post-hoc checking.

- Include in your library any model choice that could result from model checking.
- Oracle inequality ensures that super learner chooses the correct one.

Super learning and oracle inequalities

Super learning eliminates the need to put all our eggs in a single estimation basket...

...and allows us to avoid post-hoc changes to the analysis plan.

Examples of cross-validated risks (relative to linear regression) in real datasets:

Method	Study 1	Study 2	Study 3	Study 4
Linear regression	1.00	1.00	1.00	1.00
Lasso regression	0.91	0.95	1.00	0.91
D/S/A	0.22	0.95	1.04	0.43*
Ridge regression	0.96	0.90	1.02	0.98
Random forest	0.39	0.72*	1.18	0.71
MARS	0.02*	0.82	0.17*	0.61

Discrete super learner picks the best (*) estimator and fits using the full data.

Super learning and oracle inequalities

Super learning eliminates the need to bank on a single estimation technique for each of the outcome regression and propensity score.

Using the BOLD data, we estimated the outcome regression and propensity score.

- Ten algorithms were run with and without variable screening based on univariate associations.
- Table shows cross-validated risk (relative to linear/logistic regression) for each method.
- BART fit the outcome regression best, but fit the propensity score very poorly.

Method	OR	PS
GLM	1.000	1.000
Step GLM	0.998	0.999
BART	0.988*	4.462
MARS	1.016	1.010
GAM	0.995	0.995
LASSO	1.010	0.993
RF	0.997	0.991
Bayes GLM	1.000	0.998
SVM	1.082	1.013
RPART	1.129	1.041
GLM + screen	0.999	0.991
Step GLM + screen	0.999	0.992
BART + screen	0.989	4.476
MARS + screen	1.012	1.002
GAM + screen	0.994	0.987*
LASSO + screen	1.010	1.006
RF + screen	0.997	0.989
Bayes GLM + screen	0.999	0.990
SVM + screen	1.061	1.018
RPART + screen	1.129	1.059

Super learning and oracle inequalities

The discrete super learner performs as well as the best candidate regression. But what if different estimators capture different features of the data?

We may gain from taking combinations of candidate estimators rather than picking just one!

Consider estimators that are convex combinations of the candidate estimators:

$$\bar{Q}_{n,\omega} := \sum_{k=1}^{K} \omega_k \bar{Q}_{n,k} \quad \text{with } \omega_k \geqslant 0 \text{ and } \sum_{k=1}^{K} \omega_k = 1 \ .$$

This defines an infinite collection (or ensemble) of candidate estimators that contains

- each individual estimator in the library;
- each convex combination of estimators, e.g., $0.3\bar{Q}_{n,1} + 0.2\bar{Q}_{n,10} + 0.5\bar{Q}_{n,25}$.

It is straightforward to find combination that minimizes cross-validated risk

$$\omega_n^{\text{cv}} := \text{argmin}_\omega \frac{1}{V} \sum_{v=1}^{V} R_n(\bar{Q}_{v,n,\omega}) \ .$$

Super learning and oracle inequalities

Method	MSE	ω_n^{cv}	MSE	ω_n^{cv}
GLM	1.000	0.00	1.000	0.00
Step GLM	0.998	0.05	0.999	0.00
BART	0.988	0.21	4.462	0.00
MARS	1.016	0.02	1.010	0.16
GAM	0.995	0.21	0.995	0.00
LASSO	1.010	0.00	0.993	0.28
RF	0.997	0.00	0.991	0.00
Bayes GLM	1.000	0.00	0.998	0.00
SVM	1.082	0.00	1.013	0.00
RPART	1.129	0.01	1.041	0.04
GLM + screen	0.999	0.00	0.991	0.00
Step GLM + screen	0.999	0.00	0.992	0.00
BART + screen	0.989	0.00	4.476	0.00
MARS + screen	1.012	0.19	1.002	0.00
GAM + screen	0.994	0.00	0.987	0.38
LASSO + screen	1.010	0.19	1.006	0.02
RF + screen	0.997	0.00	0.989	0.11
Bayes GLM + screen	0.999	0.00	0.990	0.00
SVM + screen	1.061	0.11	1.018	0.00
RPART + screen	1.129	0.00	1.059	0.00

BOLD results for outcome regression and propensity score.

Many methods receive zero weight in the super learner.

Often, though not always, weight received is proportional to method's performance.

Super learning and oracle inequalities

```
# install and load SuperLearner package
install.packages("SuperLearner")
require(SuperLearner)

# define simple super learner library
# see listWrappers() for all algorithms included
SL.lib <- c("SL.glm","SL.step.forward","SL.mean")

# fit super learner
fit <- SuperLearner(Y = Y, X = data.frame(A,W),
                    SL.library = SL.lib,
                    method="method.CC_LS")
```

Super learning and oracle inequalities

Often, we would like to objectively evaluate the performance of the super learner.

- Theory says we do well *eventually*, but did we do well in these data?

The risk of the super learner can be estimated using cross-validation.

- First fit super learner in training sample (using nested cross-validation).
- Evaluate risk in validation sample.

Method	Study 1	Study 2	Study 3	Study 4
Linear regression	1.00	1.00	1.00	1.00
Lasso regression	0.91	0.95	1.00	0.91
D/S/A	0.22	0.95	1.04	0.43
Ridge regression	0.96	0.90	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*

Super learning and oracle inequalities

```
# load SuperLearner package
require(SuperLearner)

# define simple super learner library
SL.lib <- c("SL.glm","SL.step.forward","SL.mean")

# fit cv super learner
fit <- CV.SuperLearner(Y = Y, X = data.frame(A,W),
                       SL.library = SL.lib,
                       method="method.CC_LS")

# plot cv risk results
plot(fit)
```

Super learning and oracle inequalities

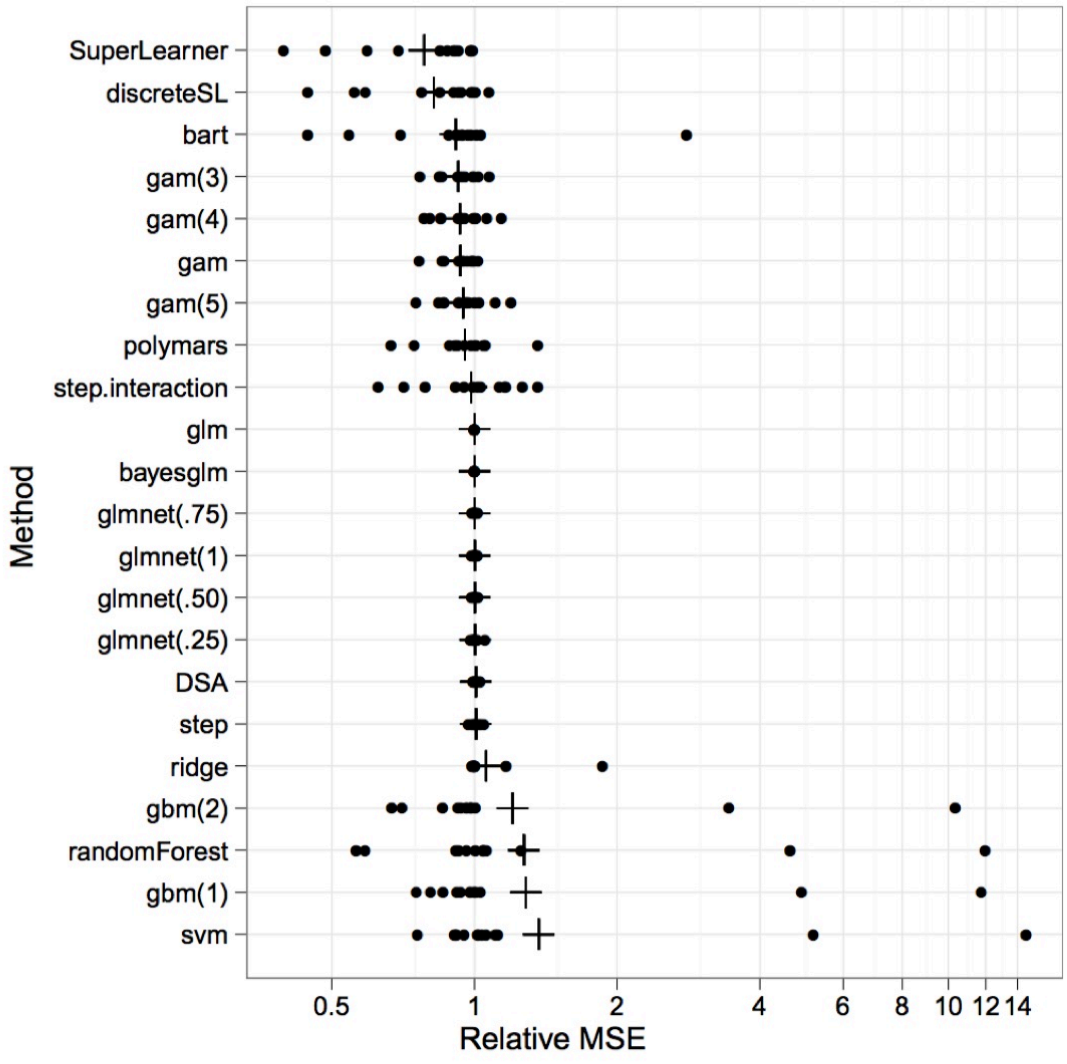
Super Learner applied to 13 public data sets (Polley, Rose, van der Laan, 2011).

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

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

Super learning and oracle inequalities

Fig. 3.4 Tenfold cross-validated relative mean squared error compared to glm across 13 real data sets. Sorted by geometric mean, denoted by the plus (+) sign



Super learning and oracle inequalities

In BOLD, super learner performed better than any individual method for both outcome regression and propensity score.

Method	OR	PS
GLM	1.000	1.000
Step GLM	0.998	0.999
BART	0.988	4.462
MARS	1.016	1.010
GAM	0.995	0.995
LASSO	1.010	0.993
RF	0.997	0.991
Bayes GLM	1.000	0.998
SVM	1.082	1.013
RPART	1.129	1.041
GLM + screen	0.999	0.991
Step GLM + screen	0.999	0.992
BART + screen	0.989	4.476
MARS + screen	1.012	1.002
GAM + screen	0.994	0.987
LASSO + screen	1.010	1.006
RF + screen	0.997	0.989
Bayes GLM + screen	0.999	0.990
SVM + screen	1.061	1.018
RPART + screen	1.129	1.059
Super learner	0.986*	0.983*

Key points of Chapter 3

- Regression is typically needed to estimate ATE.
- Borrowing information lightly leads to high bias but low variance.
- Borrowing information heavily leads to low bias but high variance.
- The risk (based on a loss function) allows us to score estimators on their bias and variance.
- Super learner uses cross-validation to choose the estimator with the best risk.
- Oracle inequalities establish that super learner is essentially as close to the truth as the oracle estimator.
- Nested cross-validation provides a useful tool for assessing the performance of the super learner itself.

References and additional reading

References:

Breiman L (1996). Stacked regressions. *Machine Learning*, 24(1)49-64. doi: 10.1023/A:1018046112532.

Pirracchio R, Petersen M, Carone M, Rigon M, Chevret S, van der Laan M. Mortality prediction in intensive care units with the Super ICU Learner Algorithm (SICULA): A population-based study (2015). *Lancet Resp Med*; 3(1):42-52. doi: 10.1016/S2213-2600(14)70239-5.

Polley E, Rose S, van der Laan M. Super Learning (2011). *Targeted Learning: Causal Inference for Observation and Experimental Data*; Chapter 3: 44-66. Springer New York. doi: 10.1007/978-1-4419-9782-1.

Rose S. Mortality risk score prediction in an elderly population using machine learning (2013). *Am J of Epi*; 177(5):443-452. doi: 10.1093/aje/kws241.

van der Laan MJ, Polley E, Hubbard A. Super learner. *Stat App Gen and Mol Bio*; 6(1). doi: 10.2202/1544-6115.1309.

van der Vaart A, Dudoit S, van der Laan MJ (2006). Oracle inequalities for multi-fold cross-validation. *Stat Decis*; 24(3)351-371. doi: 10.1524/stnd.2006.24.3.351.

Wolpert D (1992). Stacked generalization. *Neural Networks*, 5(2)241-259. doi: 10.1016/S0893-6080(05)80023-1.