ulm university universität ulm

Finance Project – Asset Management

Andre Guettler | Tim Baumgartner
Institute of Strategic Management and Finance

Course Outline

1. Motivation & Organization
2. Backtesting Fundamentals
3. Statistical Learning with Applications in R
    a. Introduction to R (Chapter 2.3) [Tim]
    b. Assessing Model Accuracy (Chapter 2.2)
    c. Linear Model Selection and Regularization (Chapter 6)
    d. Resampling Methods (Chapter 5)
4. Practical Implementation
    a. Interactive R Programming Problem Sets [Tim]
    b. Own Backtesting Strategies [Tim]

CHAPTER 1:

MOTIVATION & ORGANIZATION

Aim of the Finance Project

- Students will develop their own investment strategies based on machine learning approaches using R.
- Combination of lectures (offering key concepts in Machine Learning, Backtesting, basic R knowledge) as well as hands-on empirical implementations of your own strategies.

General Course Information I

- Lectures take place on Wednesdays at 12:15 in E60 (HeHo 18)
- See the dates on Moodle (first lecture on April 23rd)
- Then, the practical exercises start and we will offer regular consultation hours
- Important deadlines:
  - Written project report (submission by E-mail): July 1st, 6 pm
  - Presentation of final results: July 2nd

General Course Information II

- **Mandatory reading (selected chapters):**

An Introduction to Statistical Learning, James/Witten/Hastie/Tibshirani

It‘s a free eBook

Videos & Data available via:

www.dataschool.io/15-hours-of-expert-machine-learning-videos

www.statlearning.com

- **Grading**:
  - Written project report 10-15 pages using R Markdown (75%)
  - Presentation of final results (25%)
  - Work will be group-based (2-3 students)

General Course Information III

- **Slides and teaching** material will be uploaded to Moodle

- **Contact information**
  - Lecture:
    - Andre Guettler (andre.guettler@uni-ulm.de)
  - Practical implementations:
    - Tim Baumgartner (tim.baumgartner@uni-ulm.de)

CHAPTER 2:

BACKTESTING FUNDAMENTALS

Backtesting - Overview

1. Case Study
2. In-sample Tests
3. Out-of-sample Forecasts

Fact sheet – SRA Credit Spread Trading I

Investment strategy

Forecasting horizon:	1 day
Trading instruments:	$HYG ETF
Credit risk-free instrument:	US Treasury (1 year)
Benchmark:	$HYG ETF (buy & hold)
Dividend treatment:	Re-investing
Short-selling:	No
Leverage:	No
Stop-loss:	No
Rebalancing frequency:	Daily
Trading costs:	Bid/ask spreads + 2 bp ETF + 0.5 bp TSY

Key characteristics

Launch date:	May 1, 2008
Today:	February 27, 2015
AuM at launch, $	1'000.00
AuM today, $	2'141.48
# of trades:	185

Fees

Conversion	-
Management	-
Performance	-

Total returns

	Strategy	Benchmark	Outperf.
YTD	2.44%	2.96%	-0.52%
1 month	2.21%	2.21%	0.00%
3 months	1.97%	2.14%	-0.16%
6 months	-0.67%	0.06%	-0.74%
1 year	0.23%	2.17%	-1.95%
3 years (annual avg.)	4.85%	6.55%	-1.70%
5 years (annual avg.)	9.18%	9.58%	-0.40%
Incept. (annual avg.)	16.73%	7.99%	8.74%

Risk/return measures

	Since inception		Last 5 years	
	Strategy	Bench.	Strategy	Bench.
Alpha (monthly), %	0.84%	-	0.27%	-
Beta	0.15	-	0.55	-
Sharpe ratio (annual.)	1.38	0.40	1.24	0.97
Standard deviation (annual.), %	8.3%	21.0%	6.3%	8.5%
Max. draw-down, %	-7.5%	-33.5%	-4.5%	-10.6%
Max. up-turn, %	120.0%	130.0%	48.5%	51.4%
Number of up-turn days, %	55.5%	54.5%	56.6%	54.3%

Fact sheet – SRA Credit Spread Trading II

Performance, USD

Monthly strategy returns over benchmark

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Year
2015	-0.5%	0.0%											-0.5%
2014	-0.1%	0.0%	-0.5%	-0.5%	0.0%	0.0%	0.8%	-1.0%	1.2%	-1.5%	-0.3%	0.3%	-1.5%
2013	0.3%	0.0%	0.0%	-0.3%	1.9%	0.1%	-1.3%	0.3%	-0.1%	-0.4%	-0.3%	-0.5%	-0.1%
2012	-0.6%	-0.1%	-0.2%	-0.7%	0.8%	-1.5%	-0.6%	0.0%	0.4%	0.4%	-0.9%	0.0%	-3.3%
2011	0.0%	0.0%	-0.8%	0.0%	0.0%	0.6%	-0.6%	2.5%	2.5%	-1.8%	1.3%	-0.4%	3.9%
2010	2.0%	-0.9%	0.0%	-0.2%	2.1%	-0.6%	-0.6%	-0.5%	-0.5%	0.0%	0.2%	0.0%	1.3%
2009	-0.9%	9.2%	1.5%	-6.7%	-1.8%	-1.1%	0.7%	1.5%	0.0%	-3.3%	0.0%	0.0%	0.2%
2008					1.1%	3.7%	-0.4%	0.1%	10.3%	12.9%	8.5%	-5.9%	31.3%

In-sample (IS) tests – General setup I

$$r_{t,t+h} = a + bX_t + u_t \qquad (1)$$

r indicating ln(excess return)

Excess return = stock market return – risk free rate

X (vector of) predictor(s)

t equals end of period

h equals certain time span

Example for one observation:

- X_t: Dividend yield as of 31.12.2020
- $r_{t,t+6}$: Excess return for the period 31.12.2020 to 30.06.2021

In-sample tests – General setup II

- In-sample OLS estimators have high power (if model specification is valid)
- Results often depend highly on the sample period (start date and end date)
- Inference is quite tricky, in particular if you use multi-period excess returns (i.e., forecasts for six months) and if predictors are highly persistent.

In-sample tests – Inference GW 08

- Goyal and Welch (2008) uses a bootstrap following Mark (1995) and Kilian (1999)
- They construct 10,000 bootstrapped time series by drawing with replacement from the residuals.
- The initial observation—preceding the sample of data used to estimate the models—is selected by picking one date from the actual data at random.
- This bootstrap procedure preserves the autocorrelation structure of the predictor variable.

In-sample tests – Results from GW 08

In-sample adj. R squared from Welch / Goyal (2008):

Full Sample, Not Significant IS

dfy	Default yield spread	1919–2005	−1.18
infl	Inflation	1919–2005	−1.00
svar	Stock variance	1885–2005	−0.76
d/e	Dividend payout ratio	1872–2005	−0.75
lty	Long term yield	1919–2005	−0.63
tms	Term spread	1920–2005	0.16
tbl	Treasury-bill rate	1920–2005	0.34
dfr	Default return spread	1926–2005	0.40
d/p	Dividend price ratio	1872–2005	0.49
d/y	Dividend yield	1872–2005	0.91
ltr	Long term return	1926–2005	0.99
e/p	Earning price ratio	1872–2005	1.08

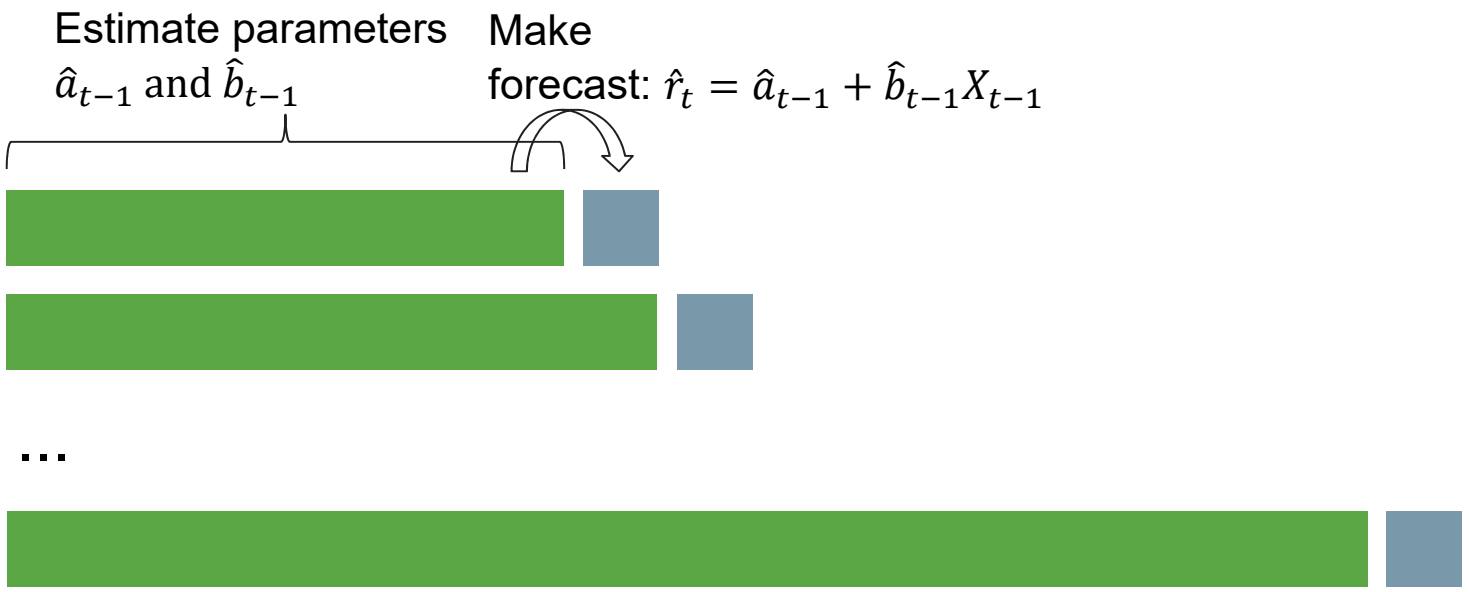
Full Sample, Significant IS

b/m	Book to market	1921–2005	**3.20***
i/k	Invstmnt capital ratio	1947–2005	**6.63****
ntis	Net equity expansion	1927–2005	**8.15*****
eqis	Pct equity issuing	1927–2005	**9.15*****
all	Kitchen sink	1927–2005	**13.81****

Welch, I., and A. Goyal (2008) "A comprehensive look at the empirical performance of equity premium prediction." Review of Financial Studies 21, 1455-1508.

Out-of-sample (OOS) tests - Overview

- OOS: regression diagnostics (robustness check for IS estimation)
- The predictive performance of a model is evaluated **outside** the sample that was used for estimating the model parameters
- Walk-forward out-of-sample test:



Out-of-sample tests

1. Compare model forecast $\hat{r}_t$ with realization r_t
2. Compare historical mean forecast $\bar{r}_t$ with realization r_t

- If we do this for all walk-forward periods, we can generate an out-of-sample R squared measure:

$$R_{OS}^2 = 1 - \frac{\sum_{t=1}^{T}(r_t - \hat{r}_t)^2}{\sum_{t=1}^{T}(r_t - \bar{r}_t)^2}$$

- Positive R squared indicate that the forecast model has a lower prediction error compared to the historical mean.

Out-of-sample results from Welch / Goyal (2008)

- In the WG 08 case, only **one** predictor (*eqis*) remain significant!

Full Sample, Not Significant IS

dfy	Default yield spread	1919–2005	−3.29
infl	Inflation	1919–2005	−4.07
svar	Stock variance	1885–2005	−27.14
d/e	Dividend payout ratio	1872–2005	−4.33
lty	Long term yield	1919–2005	−7.72
tms	Term spread	1920–2005	−2.42
tbl	Treasury-bill rate	1920–2005	−3.37
dfr	Default return spread	1926–2005	−2.16
d/p	Dividend price ratio	1872–2005	−2.06
d/y	Dividend yield	1872–2005	−1.93
ltr	Long term return	1926–2005	−11.79
e/p	Earning price ratio	1872–2005	−1.78

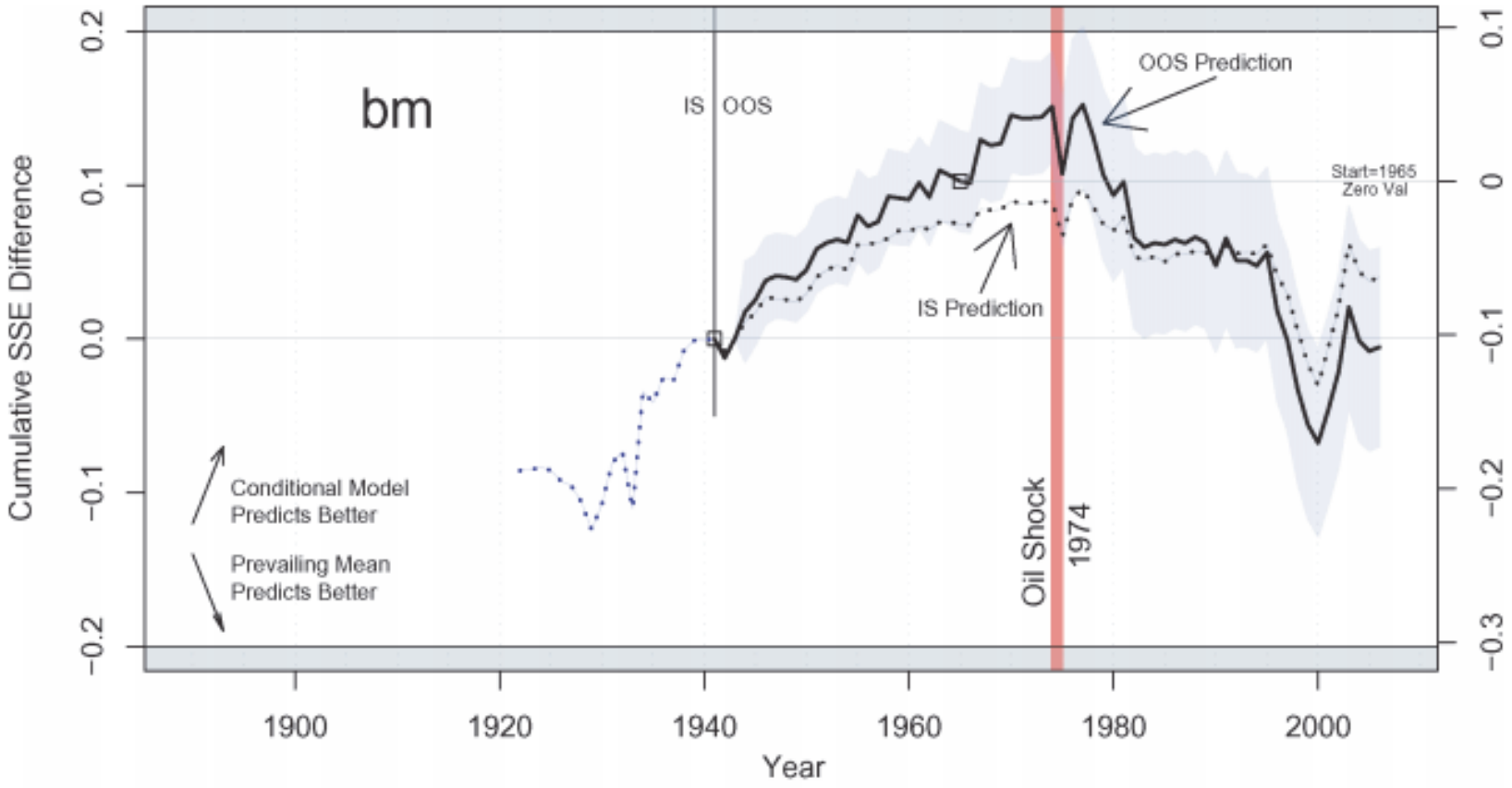
Full Sample, Significant IS

b/m	Book to market	1921–2005	−1.72
i/k	Invstmnt capital ratio	1947–2005	−1.77
ntis	Net equity expansion	1927–2005	−5.07
eqis	Pct equity issuing	1927–2005	**2.04****
all	Kitchen sink	1927–2005	−139.03

Plots

- IS: cumulative squared demeaned equity premium minus the cumulative squared regression residual.
- OOS: the cumulative squared prediction errors of the prevailing mean minus the cumulative squared prediction error of the predictive variable from the linear historical regression.
- Whenever a line increases, the predictor is gaining forecasting ability; whenever it decreases, the prevailing means (i.e., historical average) predicts better.

Plots – WG (08), Example for Book-to-Market



→ Positive IS, but negative OOS, failed the robustness check!

Useful Predictors

Requirements:

1. both significant IS **and** reasonably good OOS performance over the entire sample period
2. a generally upward drift (of course, an irregular one)
3. an upward drift which occurs not just in one short or unusual sample period (say just the two years around the Oil Shock)
4. an upward drift that remains positive over the most recent several decades (often predictors lose forecasting power after publication)

P-hacking / data mining

- Multiple testing fallacy
  - if one performs enough backtests on a single dataset, one is certain to find a backtest result that is successful to any pre-specified level of statistical significance.
- Solutions:
  - Paper performance (tracking trading performance online, increases the hurdles to do that many times in parallel for different strategies)
  - Real money performance: gold standard (many potential investors require at least three (!) years of live and reliable performance)

CHAPTER 3:

STATISTICAL LEARNING WITH APPLICATIONS IN R

3.a) Introduction to R

- Install R and download RStudio from www.rstudio.com/products/rstudio/download/#download

Roadmap:

- Usage of RStudio IDE
- R Markdown and Chunks
- Basic Commands
- Functions
- Data Structures

3.b) Assessing Model Accuracy
What is Statistical Learning?

- We observe Y and $X = (X_1, \ldots, X_p)$ for p predictors and i observations (for simplification, we do not show the i subscript).
- We believe that there is a relationship between Y (e.g., excess return of the S&P 500) and at least one of the predictors (X), e.g., Dividends.
- We can model the relationship as

$$Y = f(X) + \varepsilon$$

- Where f is an unknown function and ε is a random error term.
- Statistical learning is all about how to estimate f.
- The term statistical learning refers to using the data to “learn” f.
- In this class, we want to forecast Y using predictor(s) X

Measuring Quality of Fit

- Suppose we have a regression problem.
- One common measure of accuracy is the mean squared error (*MSE*)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Where $\hat{y}_i$ is the prediction our method gives for observation *i* in our **training data**.

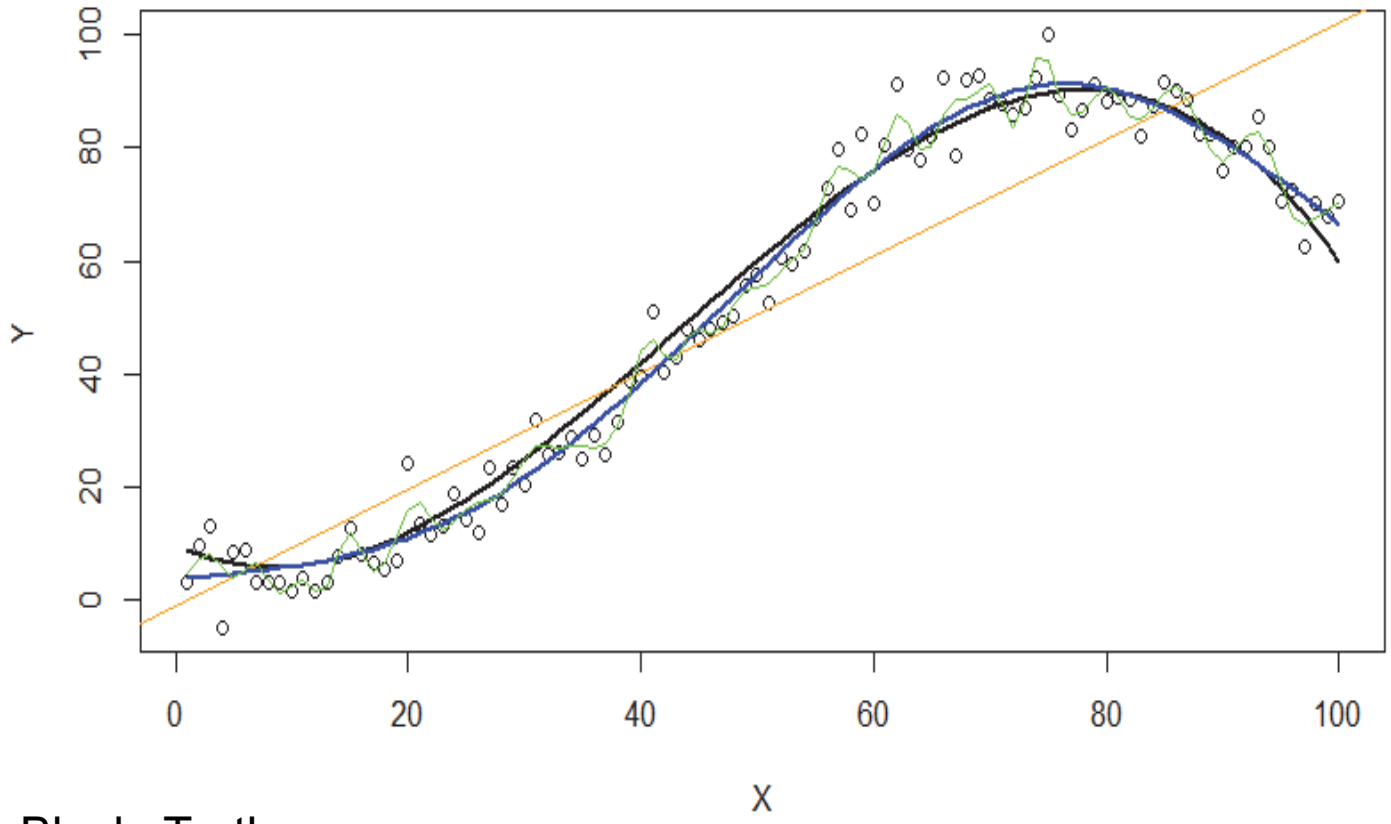
A Problem

- Our method has generally been designed to make *MSE* small on the training data we are looking at.
- For instance, with linear regression we choose the line such that *MSE* is minimized.
- What we really care about is how well the method works on new data.
- We call this new data **test data**.
- There is no guarantee that the method with the smallest training *MSE* will have the smallest test *MSE*.

Training vs. Test MSE's

- The more flexible a method is, the lower its **training *MSE*** will be, i.e., it will "fit" or explain the training data very well.
  - More flexible methods can generate a wider range of possible shapes to estimate *f* as compared to less flexible and more restrictive methods (such as linear regression). The less flexible the method, the easier to interpret the model. Thus, there is a trade-off between flexibility and model interpretability.
- However, the **test *MSE*** may in fact be higher for a more flexible method than for a simple approach like linear regression.

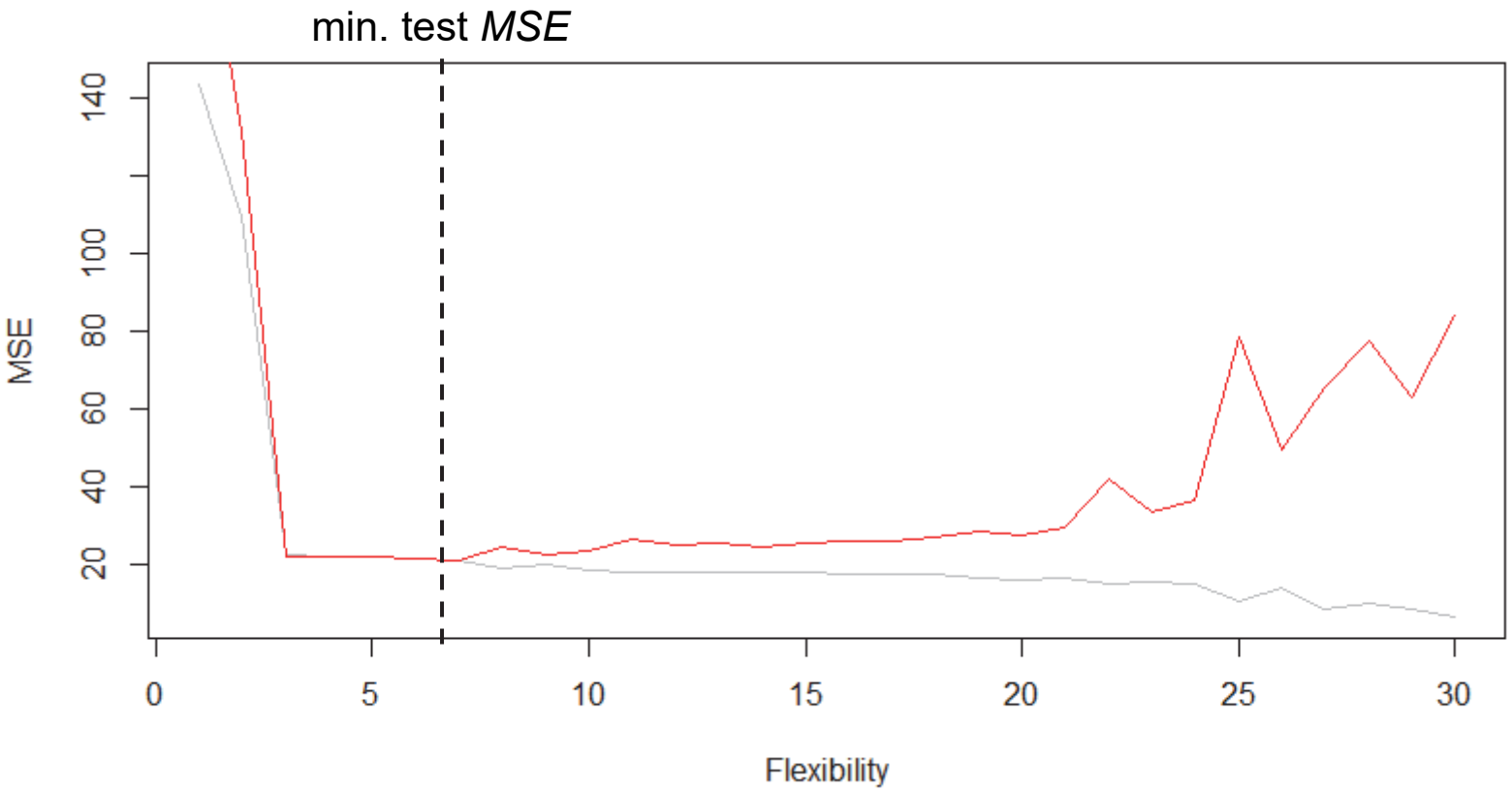
Example I



Black: Truth
Orange: OLS
Blue: Smoothing spline (less flexible)
Green: Smoothing spline (more flexible)

R-Lab:
StatLearning

Example II



Red: Test MSE
Grey: Training MSE

R-Lab:
StatLearning

Bias/Variance Tradeoff

- The previous graph of test versus training MSE's illustrates a very important tradeoff that governs the choice of statistical learning methods.

- There are always two competing forces that govern the choice of learning method, i.e., **bias** and **variance**.

Bias of Learning Methods

- Modelling (complicated) real life problems may induce an error, called bias.
- For example, linear regression assumes that there is a linear relationship between Y and X. It is unlikely that, in real life, the relationship is exactly linear so some bias will be present.
- The more flexible/complex a method is, the less bias it will generally have.

Variance of Learning Methods

- Variance refers to how much your estimate for f would change by if you had a different training data set.

- Generally, the more flexible a method is, the more variance it has.

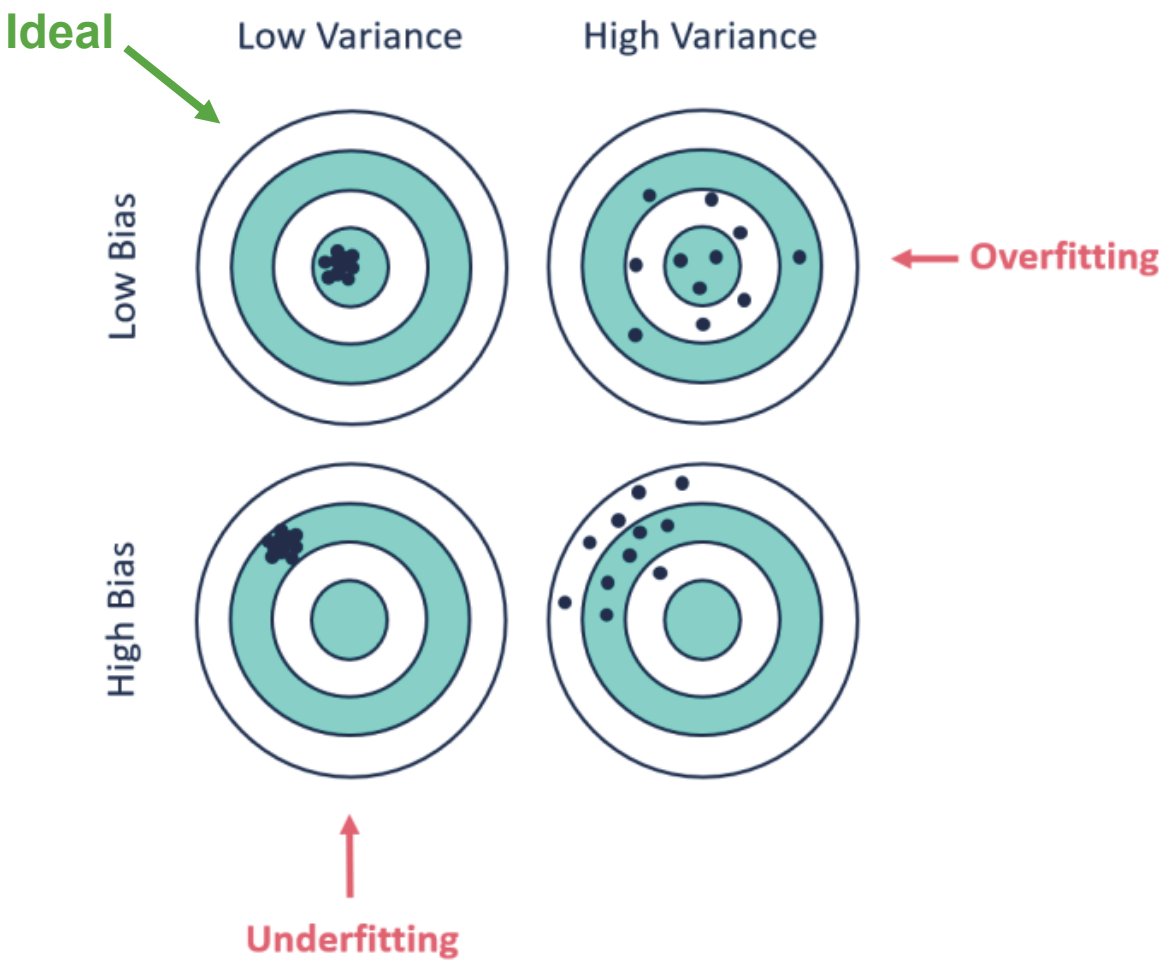
The Trade-off

- It can be shown that for any given $X=x_0$, the expected test MSE for a new Y at x_0 will be equal to

$$Expected\_Test\_MSE = E\left(Y - f(x_0)\right)^2 = Bias^2 + Var + \underbrace{\sigma^2}_{\text{Irreducible Error}}$$

- As a method gets more complex the **bias will decrease** and the **variance will increase** but expected test MSE may go up or down!

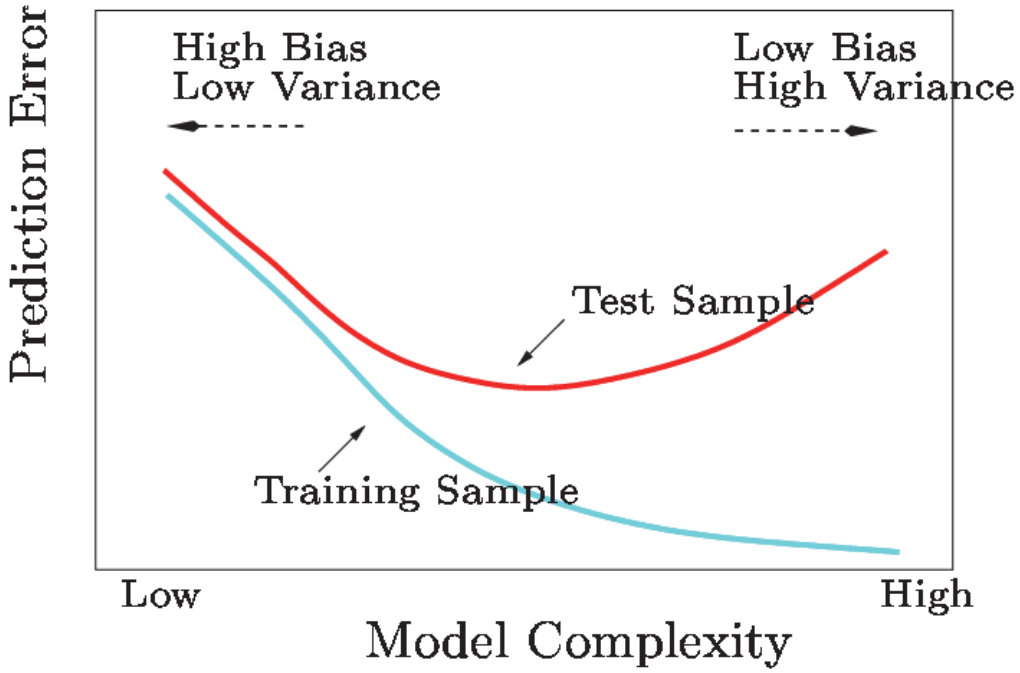
Over- versus Underfitting



Source: alteryx.com

A Fundamental Picture

- In general, training errors will always decline.
- However, test errors will decline at first (as reductions in bias dominate) but will then start to increase again (as increases in variance dominate).
- We must always keep this picture in mind when choosing a learning method.
- More flexible/complicated is not always better!



3.c) Linear Model Selection and Regularization

- OLS is the starting point

$$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

- β_0 is the intercept (i.e., the average value for *Y* if all the *X*'s are zero), β_j is the slope for the *j*th variable X_j
- β_j is the average increase in *Y* when X_j is increased by one and all other *X*'s are held constant.
- By differentiating with respect to beta, setting the equation to zero, and solving for beta, they can be obtained via

$$\beta = (X'X)^{-1}X'y.$$

- (If you have trouble with OLS, read Chapter 3 of the textbook!)

Improving on the OLS Estimates?

- We want to improve the Linear Regression model, by replacing the least square fitting with some alternative fitting procedure, i.e., the values that minimize the *MSE*.

- There are two reasons we might not prefer to just use the ordinary least squares (OLS) estimates

  1. Prediction Accuracy
  2. Model Interpretability

1. Prediction Accuracy

- The least squares estimates have relatively low bias and low variability especially when the relationship between Y and X is linear and the number of observations n is way bigger than the number of predictors p ($n>>p$).
- But, when $n \approx p$, then the least squares fit can have high variance and may result in overfitting and poor estimates on unseen observations.
- And, when $n<p$, then there is no longer a unique least squares coefficient estimate: the variance is *infinite* so the method cannot be used at all.

2. Model Interpretability

- When we have a large number of predictors, there will generally be many that have little or no effect on *Y.*
- Leaving these variables in the model makes it harder to see the “big picture”, i.e., the effect of the “important variables”.
- The model would be easier to interpret by removing (i.e., setting the coefficients to zero) the unimportant variables.
- Also, simpler models imply less information costs and faster run times.

Solutions

- Subset Selection
  - Identifying a subset of all *p* predictors *X* that we believe to be related to the response *Y*, and then fitting the model using this subset (e.g., best subset selection and stepwise selection).
- **Shrinkage** (Ridge regression and Lasso)
  - Involves shrinking the coefficient estimates towards zero.
  - This shrinkage reduces the variance.
  - Some of the coefficients may shrink to exactly zero, and hence shrinkage methods can also perform variable selection.
- Dimension Reduction (e.g., PCR)

Ridge Regression

- Ordinary Least Squares (OLS) estimates *β*'s by minimizing

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

- Ridge Regression uses a slightly different equation

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2 ,$$

Ridge Regression Adds a Penalty on *β*'s

- The effect of this equation is to add a penalty of the form

$$\lambda \sum_{j=1}^{p} \beta_j^2,$$

where the tuning parameter *λ* is a positive value.

- This has the effect of "shrinking" large values of *β*'s towards zero.
- The intercept is not subject to the penalty.
- It turns out that such a constraint should improve the fit, because shrinking the coefficients can significantly reduce their variance.
- Note that when *λ* = 0, we get the OLS.

Manual Calculation of Betas

- By differentiating with respect to beta, setting the equation to zero, and solving for beta, we obtain:

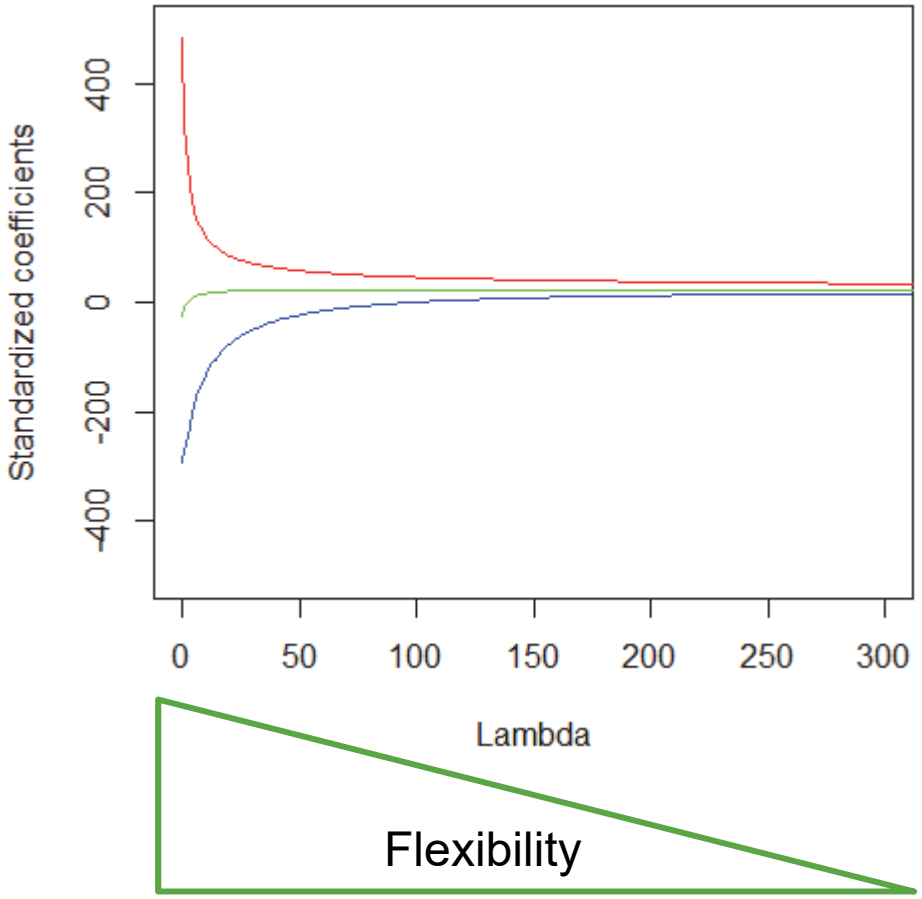
$$\beta = (X'X + \boldsymbol{\lambda I})^{-1}X'y$$

- The **penalty term** is included in parenthesis: λ times the identity matrix, I, in order to come up with a vector length that matches the number of predictors (i.e., the β vector includes β_1 to β_p).
- If the predictors are centered (i.e., have a mean of zero), the intercept β_0 equals mean(Y), hence, there is no need to include it in the above equation.

R-Lab:
Ridge_comparison

Hitters Data: Ridge Regression

- As λ increases, the standardized coefficients shrinks towards zero.



R-Lab:
Ridge_figures

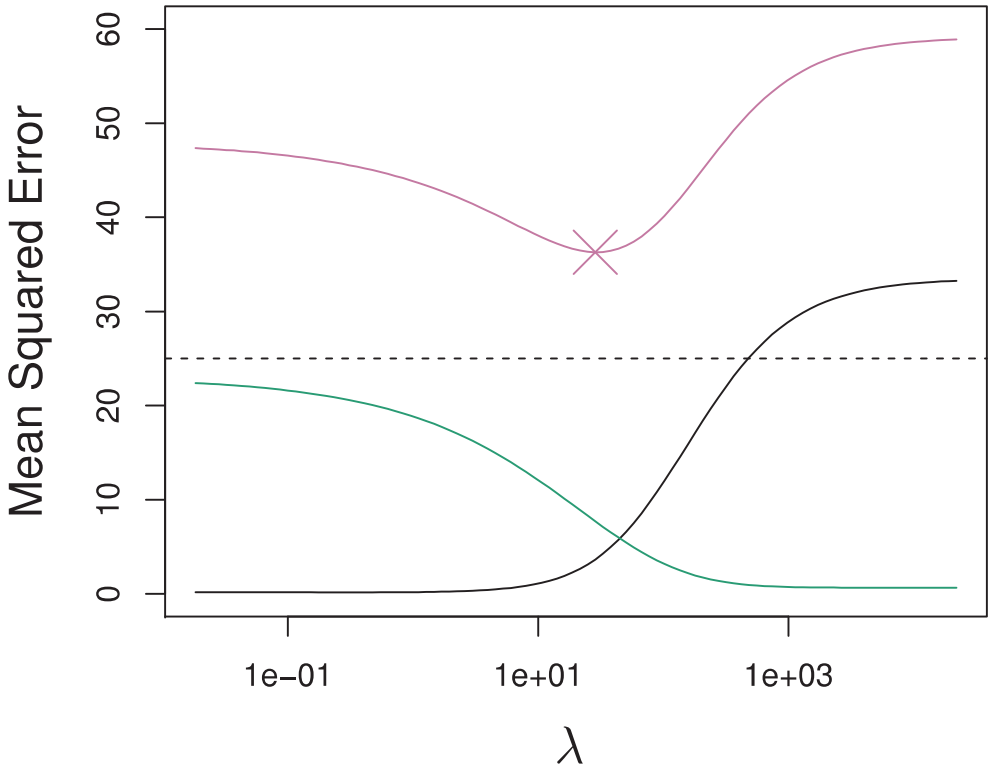
Why Can Shrinking Towards Zero Be Beneficial?

- It turns out that the OLS estimates generally have low bias but can be highly variable. In particular when n and p are of (almost) similar size, then the OLS estimates will be extremely variable

- The penalty term makes the ridge regression estimates biased, but can also substantially reduce variance.

- Thus, there is a bias/variance trade-off.

Ridge Regression Bias/ Variance Trade-off



Black: Bias

Green: Variance

Purple: Test MSE

- In general, the ridge regression estimates will be more biased than OLS estimates but have lower variance.
- Ridge regression will work best in situations where the OLS estimates have high variance.

Computational Advantages of Ridge Regression

- If p is large, then using the best subset selection approach* requires searching through enormous numbers of possible models.

  (*covered in the textbook in Section 6.1)

- With Ridge Regression, for any given λ, we only need to fit one model and the computations turn out to be very simple.

- Ridge Regression can even be used when $p > n$, a situation where OLS fails completely!

The Lasso

- One significant problem of Ridge regression is that the penalty term will never force any of the coefficients to be exactly zero.
- Thus, the final model will include all variables, which makes it harder to interpret.
- A more modern alternative is the Lasso.
- The Lasso works in a similar way to Ridge Regression, except it uses a different penalty term.

Lasso's Penalty Term

- Ridge Regression minimizes

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2,$$

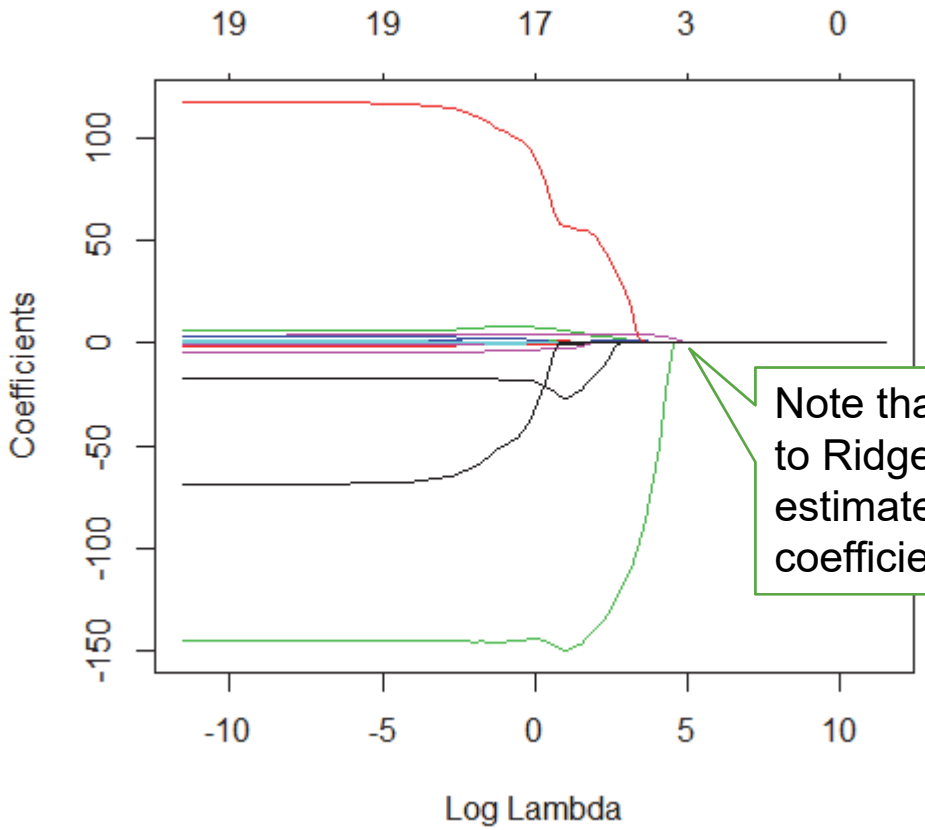
- The Lasso estimates the *β*'s by minimizing the

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|=\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|.$$

What's the Big Deal?

- This seems like a very similar idea but there is a big difference.
- Using this penalty, it could be proven mathematically that some coefficients end up being set to exactly zero.
- With Lasso, we can produce a model that has high predictive power and it is simple to interpret.
- Drawback: there is no simple closed form equation compared to the Ridge regression.

Hitters Data: Lasso



Note that, in comparison to Ridge regression estimates, standardized coefficients go to zero.

→ Now, the question arises how to select the „optimal“ λ

R-Lab: Lasso

3.d) What are Resampling Methods?

- Tools that involve **repeatedly** drawing samples from a training set and refitting a model of interest on each sample in order to obtain more information about the fitted model.
  - Model Assessment: estimate test error rates
  - Model Selection: select the appropriate level of model flexibility
- Drawback: They are computationally expensive!
- Two resampling methods:
  - **Cross Validation**
  - Bootstrapping (not used in this class)

Cross Validation

- In this course, we cover three different types of **cross validation**:
  - The Validation Set Approach
  - Leave-One-Out Cross Validation
  - K-fold Cross Validation

Typical Approach: The Validation Set Approach

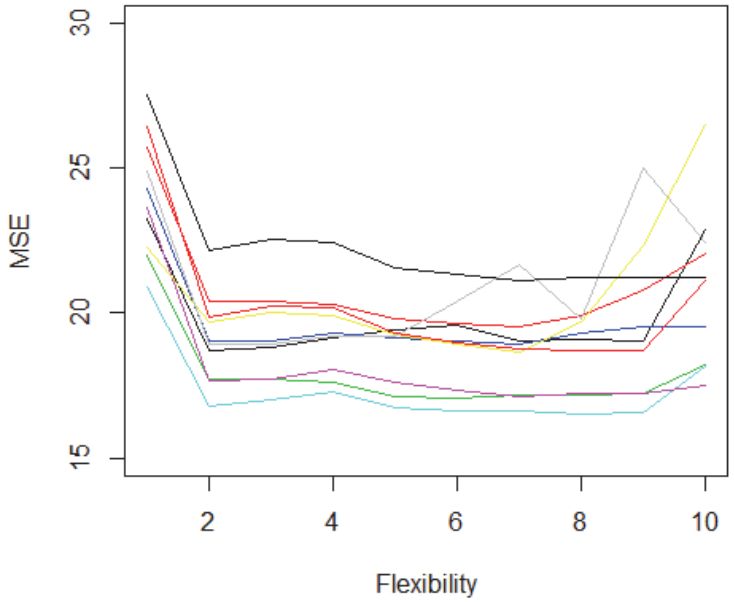
- Suppose that we would like to find a set of variables that give the lowest test (not training) error rate.
- If we have a large data set, we can achieve this goal by randomly splitting the data into training and validation (testing) parts.
- We would then use the training part to build each possible model (i.e., the different combinations of variables) and choose the model that gave the lowest error rate when applied to the validation data.

Example: Auto Data

- Suppose that we want to predict **mpg** from **horsepower.**
- Two models:
  - mpg ~ horsepower
  - mpg ~ horsepower + horsepower^2 (+ higher order polynomials)
- Which model gives a better fit?
  - Randomly split **Auto** data set into training (196 obs.) and validation data (196 obs.).
  - Fit both models using the training data set.
  - Then, evaluate both models using the validation data set.
  - The model with the lowest estimated test MSE is the winner!

Results: Auto Data

- Validation method repeated 10 times, each time the split is done randomly!
- There is a lot of variability among the MSE's… Not good! We need more stable methods!



R-Lab: ValSetApp

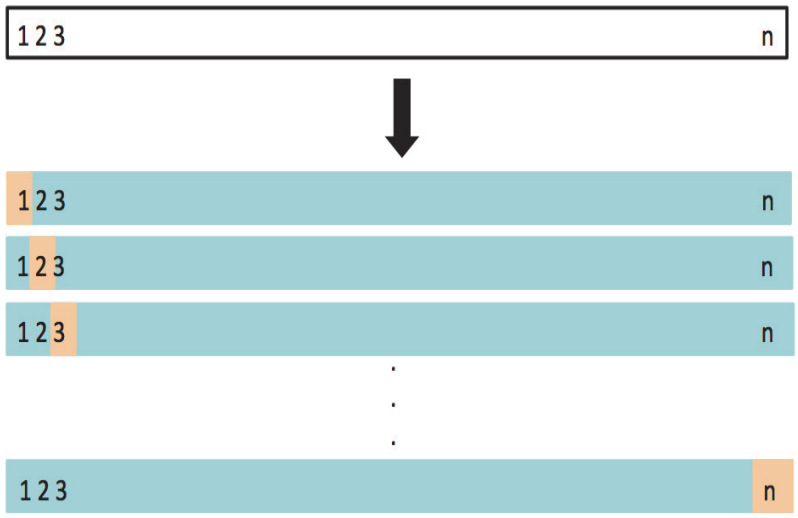
The Validation Set Approach

- Advantages:
  - Simple
  - Easy to implement

- Disadvantages:
  - The validation MSE can be highly variable.
  - Only a subset of observations are used to fit the model (training data). Statistical methods tend to perform worse when trained on fewer observations.

Leave-One-Out Cross Validation (LOOCV)

- This method is similar to the Validation Set Approach, but it tries to address the latter's disadvantages.
- For each suggested model:
  - Split the data set of size n into
    - Training data set (blue) size: n -1
    - Validation data set (beige) size: 1
  - Fit the model using the training data
  - Validate model using the validation data, and compute the corresponding MSE
  - Repeat this process n times
  - The MSE for the model is computed as follows: $CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i$.



LOOCV vs. the Validation Set Approach

- LOOCV has less bias.
  - We repeatedly fit the statistical learning method using training data that contains n-1 obs., i.e., almost all the data set is used.
- LOOCV produces a less variable MSE.
  - The validation approach produces different MSE when applied repeatedly due to randomness in the splitting process.
  - Performing LOOCV multiple times will always yield the same results, because we split based on one observation each time.
- LOOCV is computationally intensive (disadvantage).
  - We fit each model n times!

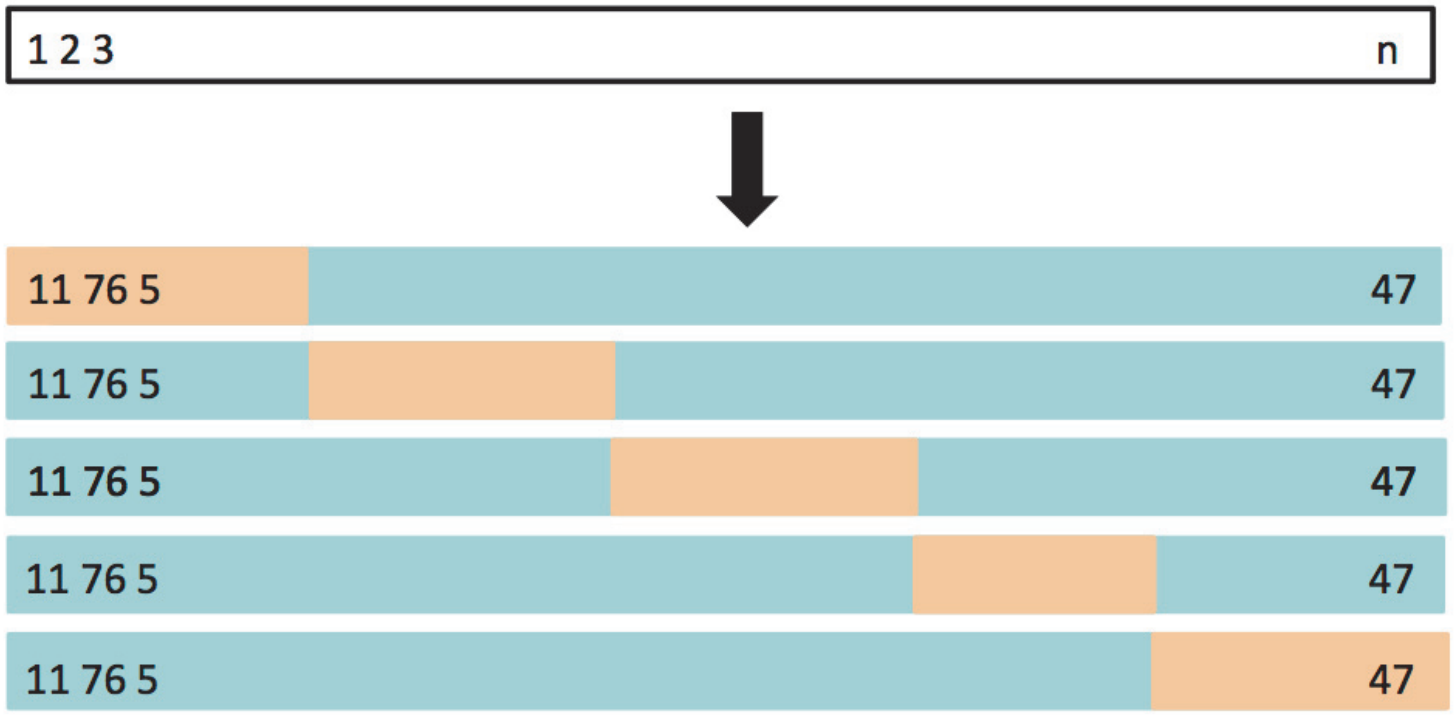
R-Lab: LOOCV

K-fold Cross Validation

- LOOCV is computationally intensive, so we can run K-fold Cross Validation instead.
- With K-fold Cross Validation, we divide the data set into K different parts (e.g. K = 5, or K = 10, etc.).
- We then remove the first part, fit the model on the remaining K-1 parts, and see how good the predictions are on the omitted part (i.e., compute the MSE on the first part).
- We then repeat this K different times taking out a different part each time
- By averaging the K different MSE's we get an estimated test error rate for new observations.

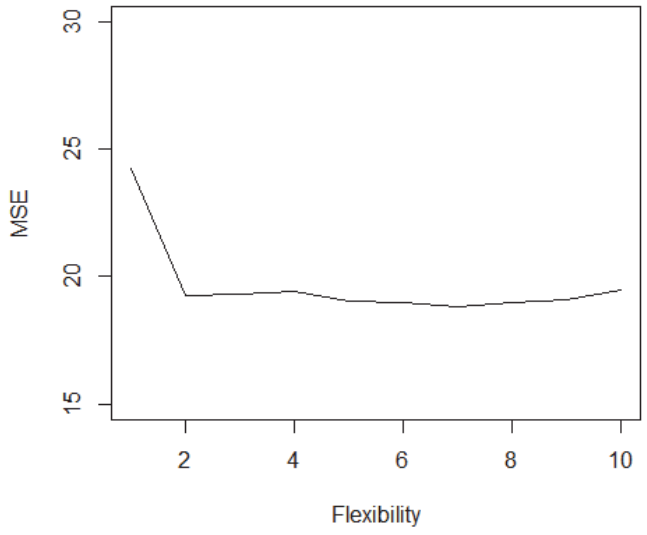
$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i .$$

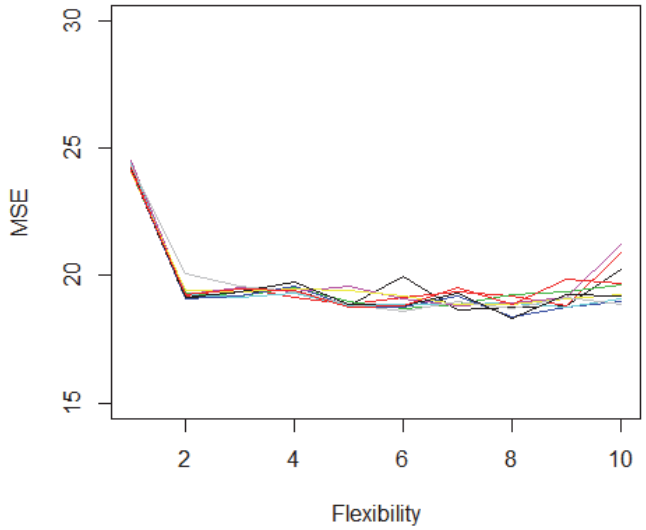
K-fold Cross Validation



Auto Data: LOOCV vs. K-fold CV

- Left: LOOCV error curve
- Right: 5-fold CV was run 10 times, and the figure shows the slightly different CV error rates.
- LOOCV is a special case of K-fold, where k = n
- They are both stable, but LOOCV is more computationally intensive!





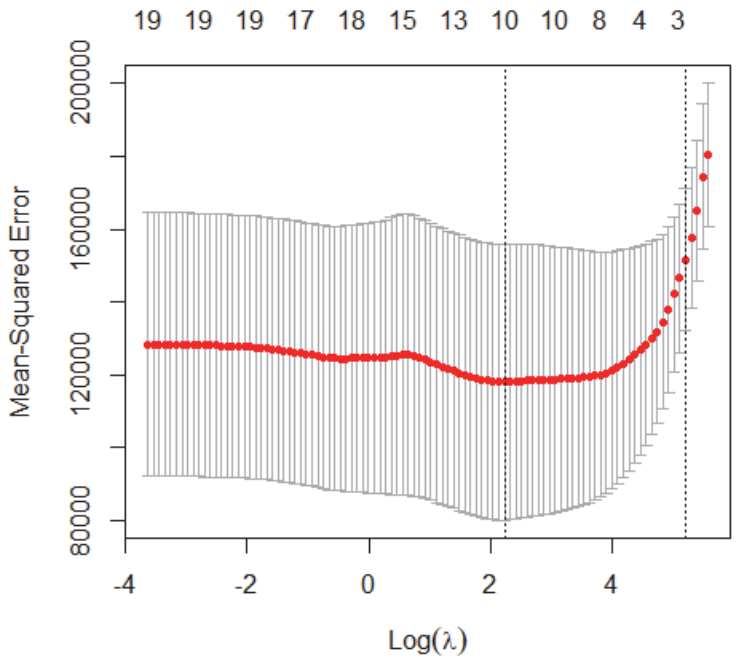
R-Labs:
LOOCV
5fold_CV

What Do We Do in Practice?

- We tend to use K-fold CV with (K = 5 and K = 10)
- It has been empirically shown that these parameters yield test error rates that suffer neither from excessively high bias, nor from very high variance.

Lasso: Selecting the Tuning Parameter λ

- We need to decide on a value for λ
- Select a grid of potential values, use **cross validation** to estimate the error rate on test data (for each value of λ), and select the value that gives the smallest error rate.



In this example, the minimum MSE is achieved at around 9.3 (log(2.2)). Only 10 out of 19 coefficients remain in the model.
The other 9 have been shrunk to zero.

R-Lab: Lasso

OLS post-Lasso

- Although Lasso's penalty term mitigates overfitting the data by producing a sparse solution, it also tends to shrink the coefficients for the selected variables too much.
- OLS post-Lasso
  1. First use the Lasso to reduce the dimension of the model
  2. To lessen the biases in the Lasso coefficient estimates, the coefficients for the selected predictor variables are re-estimated using OLS.
  3. Standard errors need to be adjusted!

R-Lab:
OLS_Post_Lasso

Elastic Net

- Elastic net approach combines Ridge and Lasso.
- In the glmnet package, the parameter α defines the mix between Ridge (0) and Lasso (1).
- It is useful to think of α as controlling the mixing between the two penalties (from Ridge and Lasso), and λ controlling the amount of penalization.
- For the Hitters data set, Ridge regression ($\alpha = 0$) yield the lowest test MSE.

R-Lab:
EN

CHAPTER 4:

PRACTICAL IMPLEMENTATION