

# Introduction to Statistical Data Science

Dr. Francois-Xavier Briol  
Department of Statistical Science,  
UCL

# Outline

- This is a selection of regression methods which I believe is the core of practical, applied statistics beyond the widespread linear/GLM framework.
- Like the exposure in linear/GLMs, this is from the point of view of not only providing black-boxes, but also understanding their properties.
- Little of what follows is cutting-edge research in Statistics itself. Nevertheless, these ideas are fundamental enough that still form the basis of much on-going methodological research.

# **BEST SUBSET SELECTION**

# Model Selection, and Motivation

- Recall the linear model

$$Y = \beta_0 + \beta_1 x_1 + \dots \beta_p x_p + \epsilon$$

- We might be interested in:
  - Improving **prediction**: are my estimates unreliable to the point that I could actually do better by removing some of the covariates?
  - Improving **interpretability**: does it make sense to include covariates for which I have no evidence that they contribute to explaining  $Y$ ?

# Sparsity

- A model is sparse if many of its components are “inactive”. In our context, this will mean several regression coefficients are zero.
- We do not need to believe that Nature is sparse in order to agree that there are advantages on having sparse *estimates*.
- Ockham’s razor: “The simplest solution is most likely the right one”.

# Strategies

- **Combinatorial search** (a.k.a subset selection): search among possible subsets of inputs, fit final model based on the chosen subset.
- **Shrinkage** (or regularisation): change your fitting criterion to “incentivize” parameters to be zero, or near-zero.
  - We will exploit this more in the context of high-dimensional regression.
- **Dimensionality reduction**: transform your inputs to a smaller set of features, regress on them. More on that in Week 8 (see also, the Machine Learning course)

# Combinatorial Search

- Core idea: given your original set of  $p$  predictors, propose several subsets of it.
  - Concern: how many subsets are out there?
- Score: for which subset, provide a quantification of how good/bad they are.
  - Concern: take the residual sums of squares (RSS). Take two sets  $S_1$  and  $S_2$  such that  $S_1 \subset S_2$ . Which one is best according to the RSS?
  - Recall the Exercise Sheet from a couple of weeks ago...

# The Best Subset Selection Algorithm

---

**Algorithm 6.1** *Best subset selection*

---

1. Let  $\mathcal{M}_0$  denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
  2. For  $k = 1, 2, \dots, p$ :
    - (a) Fit all  $\binom{p}{k}$  models that contain exactly  $k$  predictors.
    - (b) Pick the best among these  $\binom{p}{k}$  models, and call it  $\mathcal{M}_k$ . Here *best* is defined as having the smallest RSS, or equivalently largest  $R^2$ .
  3. Select a single best model from among  $\mathcal{M}_0, \dots, \mathcal{M}_p$  using cross-validated prediction error,  $C_p$  (AIC), BIC, or adjusted  $R^2$ .
- 

(ISLR, Chapter 6)




# Combinatorial Search

- For the computation, we will need **heuristics**, ways of cutting down the search
  - No guarantees of optimality, unless restrictive assumptions are used.
- For the scoring, we will follow this general idea (written as functions to be *minimised*):

$$Score = -fitness + complexity\ penalty$$

Increases by how much residuals  
are minimised



Increases by how many parameters are used



# The Best Subset Selection Algorithm

---

## Algorithm 6.1 *Best subset selection*

---

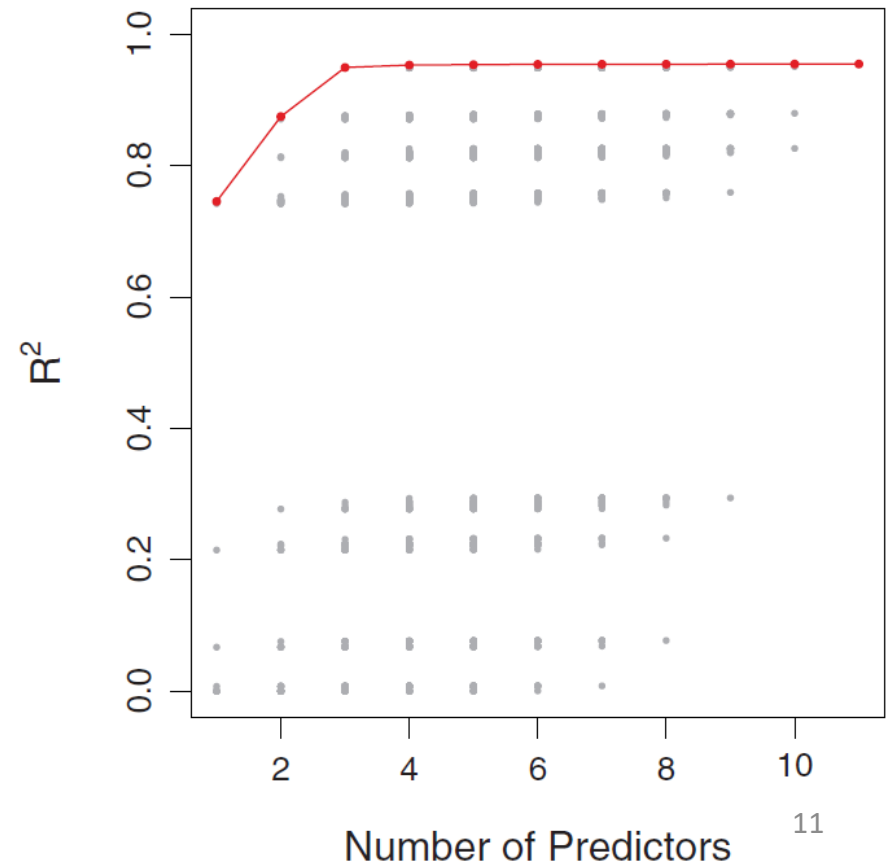
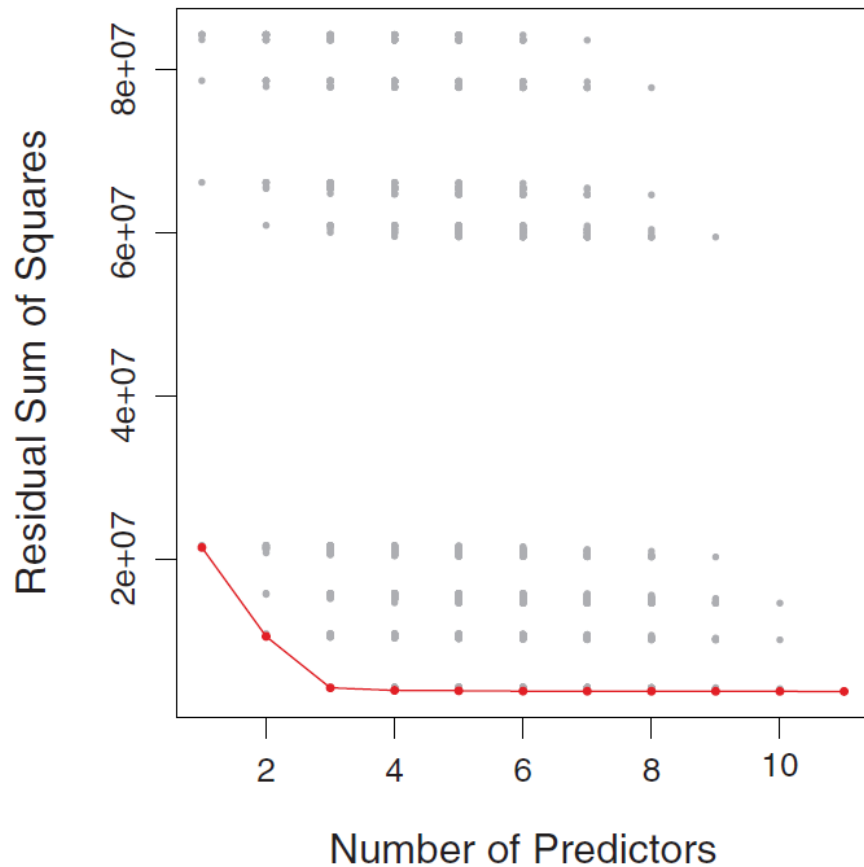
1. Let  $\mathcal{M}_0$  denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
  2. For  $k = 1, 2, \dots, p$ :
    - (a) Fit all  $\binom{p}{k}$  models that contain exactly  $k$  predictors.
    - (b) Pick the best among these  $\binom{p}{k}$  models, and call it  $\mathcal{M}_k$ . Here *best* is defined as having the smallest RSS, or equivalently largest  $R^2$ .
  3. Select a single best model from among  $\mathcal{M}_0, \dots, \mathcal{M}_p$  using cross-validated prediction error,  $C_p$  (AIC), BIC, or adjusted  $R^2$ .
- 

(ISLR, Chapter 6)

# Illustration

- *Credit* dataset: modelling balance.

(ISLR, Chapter 6)



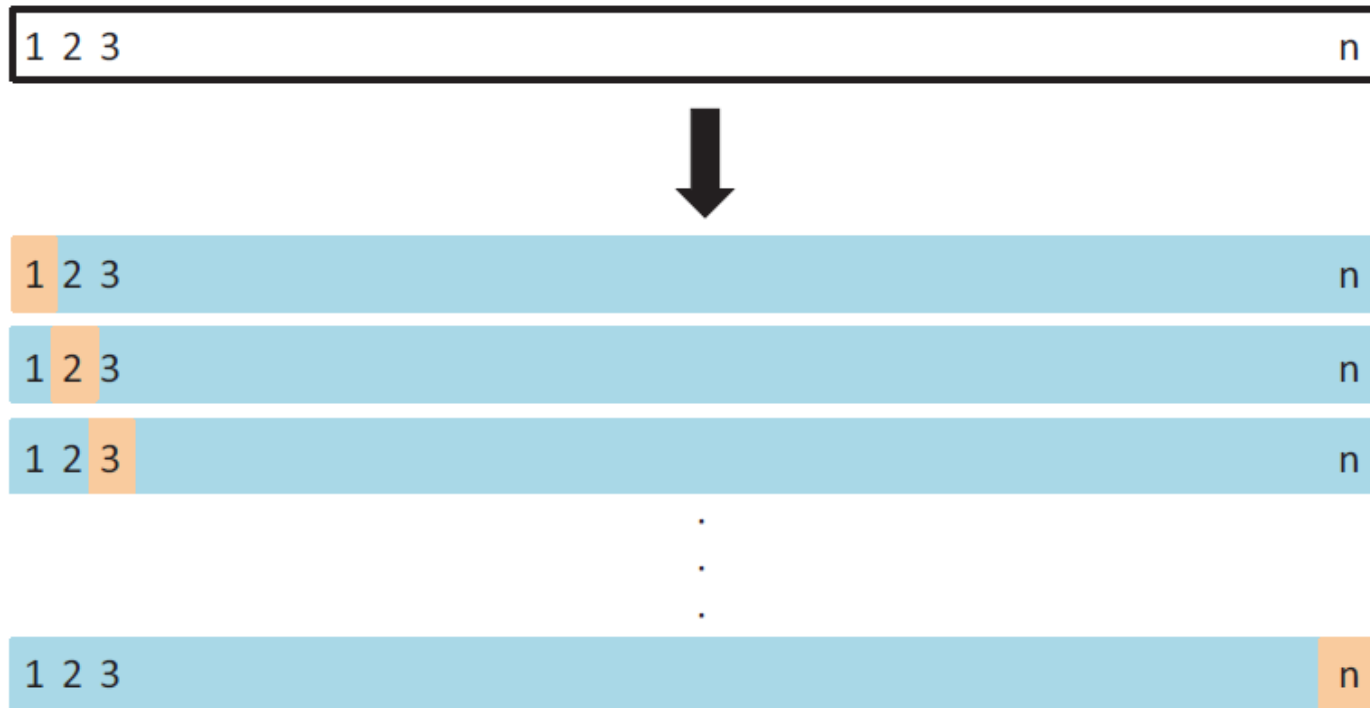
# **CRITERIA FOR SUBSET SELECTION**

# Different Criteria

- Cross-validation:
  - Split your data in  $K$  disjoint subsets.
  - For every data subset  $D_k$ , where  $k = 1, 2, \dots, K$ , use the remaining data for fitting (training) the model, and assess the RSS on the “test set”  $D_k$ .
  - Report as a score the model average  $D_k$  across folds.
- Notice that the penalty is implicit here.

# Cross-Validation

- Illustration of leave-one-out CV (LOOCV)



(ISLR, Chapter 5)

# Mallow's $C_p$

Training RSS with input set  $S$

Estimate of error variance using all covariates

$$C_p(S) \equiv \frac{1}{n} (RSS(S) + 2|S|\hat{\sigma}^2)$$

Number of covariates

- Provides an estimate of test error.
  - In practice, for linear regression it provides a similar estimate as cross-validation, without the need to split the data
  - Unlike cross-validation, not easily adaptable outside linear regression.

# AIC/BIC/\*IC

- There is a whole industry of defining penalized fitness using likelihoods: “information criteria”.
  - In our context, we use Gaussianity assumptions for the error terms.
- Unlike cross-validation and  $C_p$ , they focus on model fitness (as given by the likelihood) instead of prediction as pre-specified by a cost function, like least-squares.
  - Sometimes these measures agree, but not always.
  - If the main goal is prediction, cross-validation is in general more appealing, but it is expensive and more unstable (as it depends on the data split)



# AIC

- The AIC (Akaike Information Criterion):

$$AIC(S) \equiv -2l(S) + 2|S|$$

Log likelihood using all data

(Please notice that in some books, you might find different but equivalent definitions)

- It is not hard to show that, for linear Gaussian models, AIC and Mallow's  $C_p$  give exactly the same ranking of models.

# BIC (Bayesian Information Criterion)

$$BIC(S) = -l(S) + \frac{|S|}{2} \log(n)$$

- Which to use? Both methods start from different assumptions, which are technical and won't be discussed.
- In practice, for linear models fit by least-squares or MLE, any of these will do reasonably OK. BIC adds a stronger penalty than AIC /  $C_p$ , so it will generate smaller models. AIC might still get better out-of-sample prediction errors.

# **SCALABLE SUBSECT SELECTION**

# Greedy Search

- In combinatorial optimisation, a common heuristic is **greedy search**:
  1. Start with a (possibly random) initial candidate model
  2. Look at its “neighbours”. If any of the neighbours is better than the current candidate, make it the current candidate. Else, stop
  3. Return to Step 2

# Greedy Search

- A “starting model” can be the model with the intercept only, or a model with all covariates.
- In the context of model selection for regression models, a “neighbour” is a model that differs from the current one by having one more or one fewer covariate.
- Even simpler, depending on the starting point, we may decide only to add, or only to exclude, a covariate to/from the current candidate.
- This is commonly known as **stepwise selection**.

# Forward Stepwise Selection

---

**Algorithm 6.2** *Forward stepwise selection*

---

1. Let  $\mathcal{M}_0$  denote the *null* model, which contains no predictors.
  2. For  $k = 0, \dots, p - 1$ :
    - (a) Consider all  $p - k$  models that augment the predictors in  $\mathcal{M}_k$  with one additional predictor.
    - (b) Choose the *best* among these  $p - k$  models, and call it  $\mathcal{M}_{k+1}$ . Here *best* is defined as having smallest RSS or highest  $R^2$ .
  3. Select a single best model from among  $\mathcal{M}_0, \dots, \mathcal{M}_p$  using cross-validated prediction error,  $C_p$  (AIC), BIC, or adjusted  $R^2$ .
-

# “Local” Minimum

- Just like as in standard (continuous) optimisation, it is possible to get trapped into a “local” minimum in a combinatorial optimisation problem.
  - “Local” here is defined with respect to the neighbourhood definition.
- Notice one aspect of the particular stepwise approach from the previous slide: its stopping criterion is “follow a path until no more variables can be added. Only then decide on the best.”

# Computational Complexity

- Recall: original exhaustive search (also known as a **brute-force** approach) required the evaluation of  $2^p$  models.
- The search space of stepwise selection here required  $O(p^2)$  evaluations (why?).



(the “O” notation roughly means “something proportional to  $p^2$ ”, up to multiplicative and additive constants)



# Illustration

- *Credit* data
  - (also, R demo)

# Variables	Best subset	Forward stepwise
One	<code>rating</code>	<code>rating</code>
Two	<code>rating, income</code>	<code>rating, income</code>
Three	<code>rating, income, student</code>	<code>rating, income, student</code>
Four	<code>cards, income, student, limit</code>	<code>rating, income, student, limit</code>

# Backward Selection

---

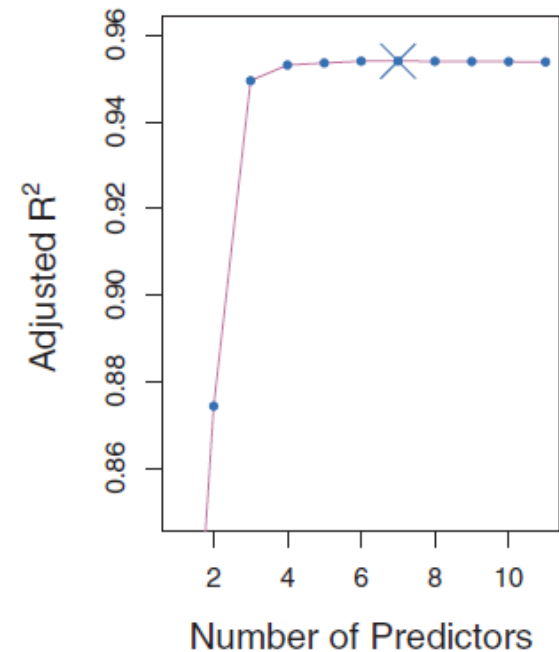
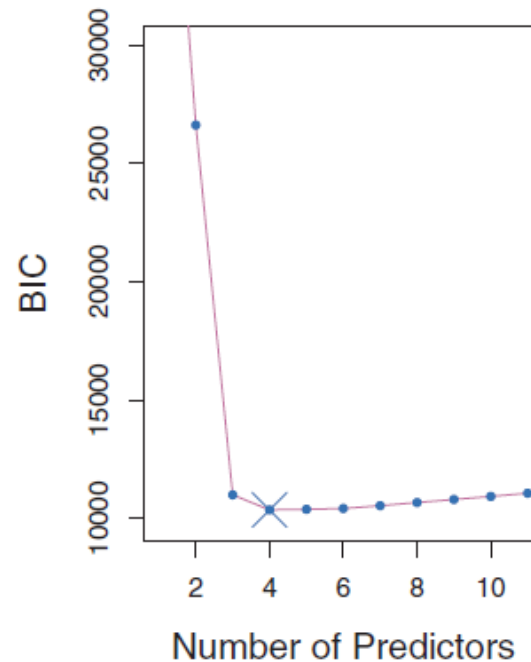
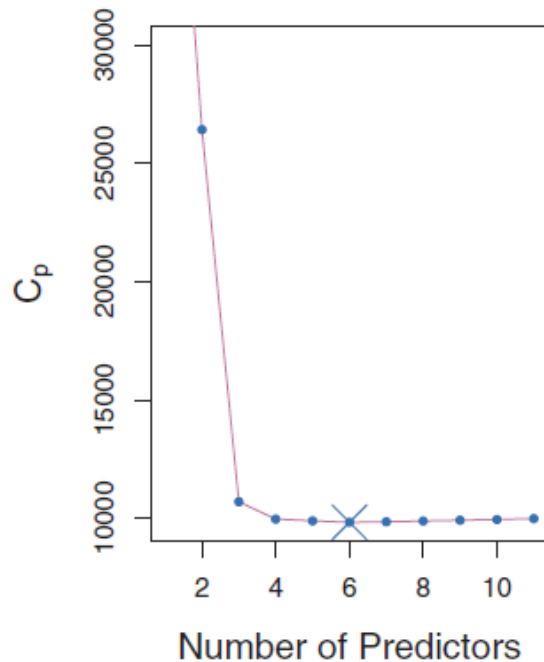
**Algorithm 6.3** *Backward stepwise selection*

---

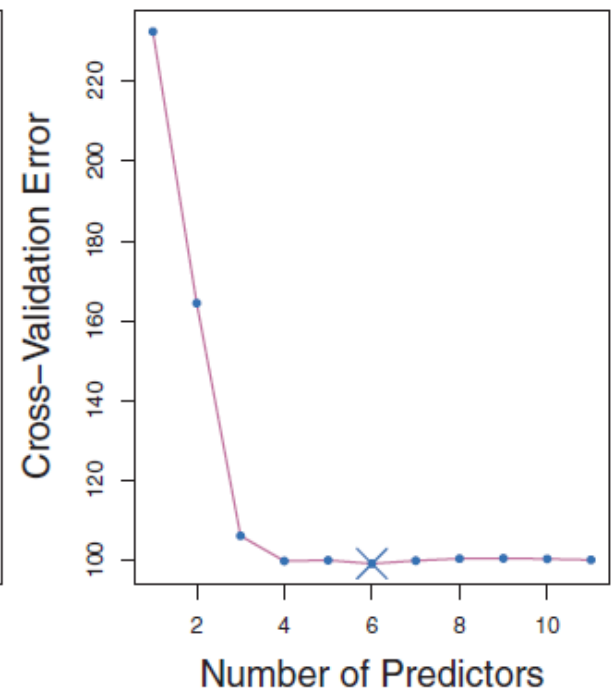
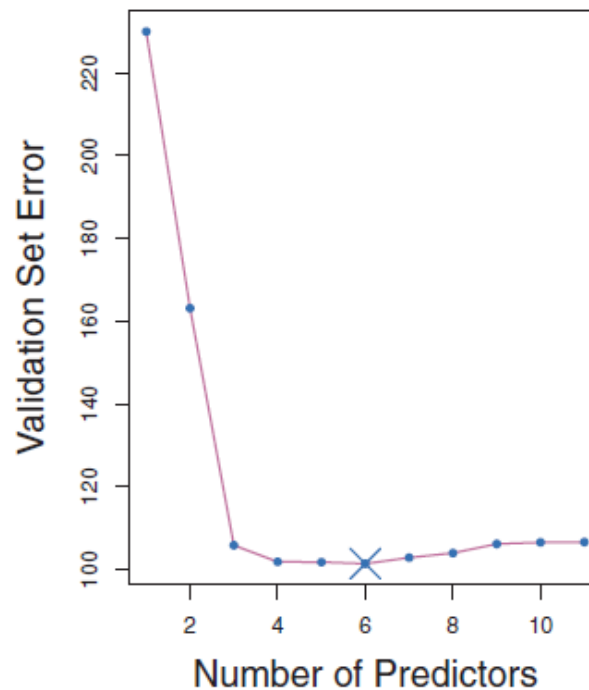
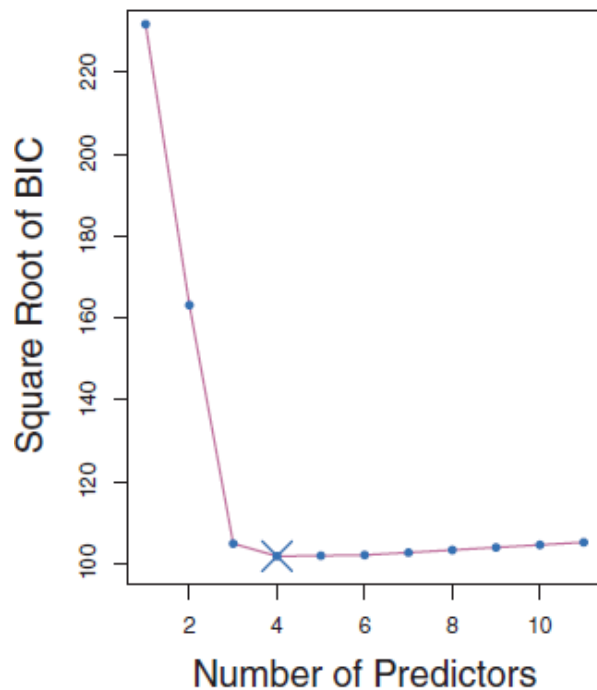
1. Let  $\mathcal{M}_p$  denote the *full* model, which contains all  $p$  predictors.
  2. For  $k = p, p - 1, \dots, 1$ :
    - (a) Consider all  $k$  models that contain all but one of the predictors in  $\mathcal{M}_k$ , for a total of  $k - 1$  predictors.
    - (b) Choose the *best* among these  $k$  models, and call it  $\mathcal{M}_{k-1}$ . Here *best* is defined as having smallest RSS or highest  $R^2$ .
  3. Select a single best model from among  $\mathcal{M}_0, \dots, \mathcal{M}_p$  using cross-validated prediction error,  $C_p$  (AIC), BIC, or adjusted  $R^2$ .
- 

(R demo: different starting points)

# Different Criteria



# Held-out Data Criteria



# Rules of Thumb

- Remember: **results in a test set are statistical estimates, not statements of fact about future out-of-sample behaviour.**

# Questions

- “Once I’ve selected the model, can I interpret the p-values of the coefficients as before?”
  - No, for the same reason that training error is not a good estimate of out-of-sample performance.
  - You could evaluate p-values in a separate test data, although your sample size will be reduced, and as a consequence so will your power.

# More Questions

- “Do I have theoretical guarantees of optimal variable selection (assuming “true sparsity” holds)?”
  - Up to some point (conditions apply), if you were able to compute the optimal solution to the combinatorial search problem. But in general this is not possible. Sub-optimal selection can still be achieved.

# Even More Questions

- “Do I have theoretical guarantees of improved test set error?”
  - Again, up to some point and with fewer assumptions (more so with cross-validation than with other methods).
  - It might not be optimal, but you can still get an improvement.



# Final Note: The p-Value Route

- Why not just drop the covariates with a corresponding low p-value (maybe with some Bonferroni correction)?
- Recall the shortcoming: each coefficient  $\beta_i$  is what we learn about the contribution of  $x_i$  **when we fix all other covariates**.
- When we drop *one* covariate, it is possible that previously non-significant coefficients now become significant.

# Final Note: The p-Value Route

- As a heuristic, it might be a cheap alternative (R demo: see what happens with the *Credit* data), but it also has no guarantees of optimality (without further assumptions).
- And it is even less clear what it would mean in terms of prediction quality.
- What about using it as the greedy search criterion?
  - Possible, but how to set the threshold of significance?

# **PENALISED REGRESSION WITH RIDGE REGRESSION**

# Regression with Limited Data

- As the number  $p$  of covariates increases, the greedy search becomes less reliable. As a matter of fact, even least-squares gets less reliable (meaning that confidence intervals get wider and wider).
- Also, in some applications we may even have situations where  $p > n$ . Least-squares is not even defined in that case.
- Let's tackle these issues using the sledgehammer of regularization, or **shrinkage**.

# Complexity as “Parameter Size”

- Consider this modified objective function for least-squares regression: ***ridge regression***

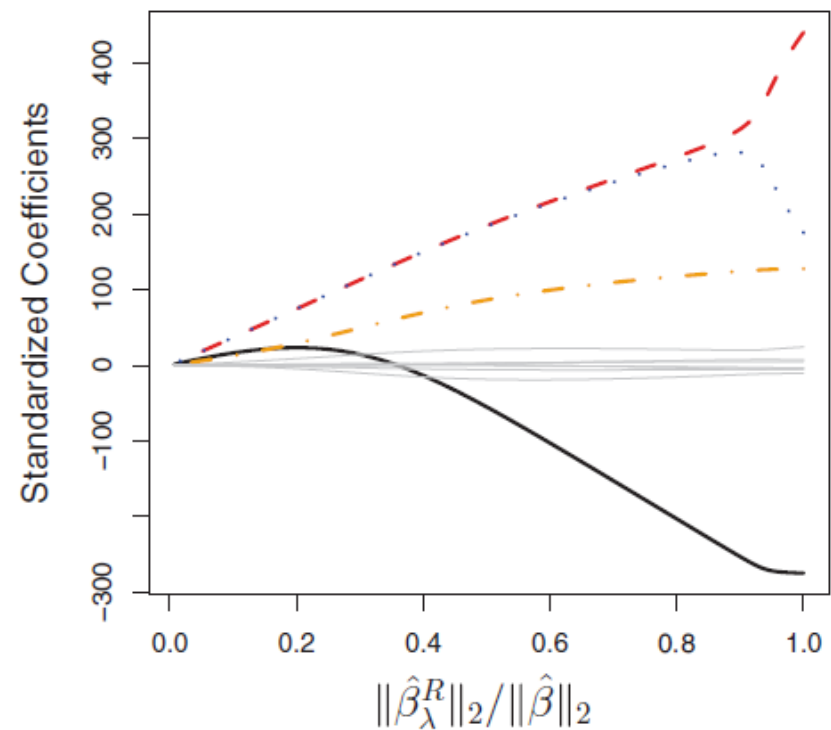
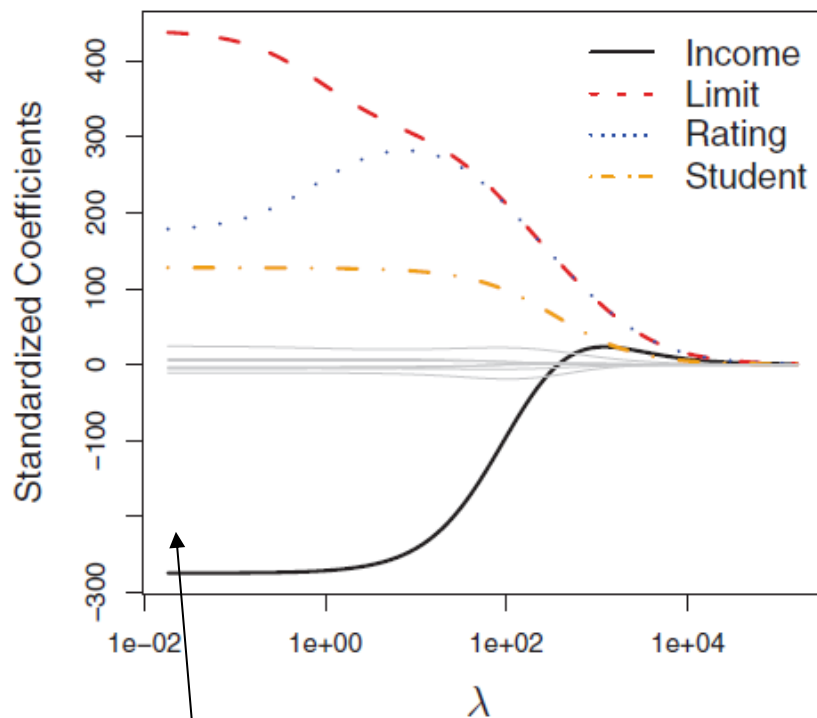
$$\sum_{i=1}^n \left( y^{(i)} - \beta_0 - \sum_{j=1}^p \beta_j x_j^{(i)} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

Our old friend, the RSS:  
measure of fit

Penalty term:  
to be set by cross-validation

Increases with “size” of  
parameters

# Illustration: *Credit* dataset



Least-squares/linear  
regression

$$\|\beta\|_2 \equiv \sqrt{\sum_{j=1}^p \beta_j^2}$$

# IMPORTANT!

- Unlike unpenalized least-squares regression, regression with shrinkage is **scale-dependent**.
- Running penalized least-squares without considering the format of your data might give you suboptimal results.
- Consider standardization:



$$\tilde{x}_j^{(i)} \equiv \frac{x_j^{(i)}}{S_j}, \text{ where } S_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_j^{(i)} - \bar{x}_j)^2}$$

**R demo: dangers of ignoring data scale in ridge regression.**

# More Intuition

- The **bias-variance trade-off**, mean-squared error style:

$$\begin{aligned}MSE &= E[(\theta - \hat{\theta})^2] = \\&= E[\theta^2] - 2E[\theta\hat{\theta}] + E[\hat{\theta}^2] \\&= \theta^2 - 2\theta E[\hat{\theta}] + E[\hat{\theta}^2] \\&= \theta^2 - 2\theta E[\hat{\theta}] + \color{red}{E[\hat{\theta}]^2} - \color{red}{E[\hat{\theta}]^2} + E[\hat{\theta}^2] \\&= (\theta - E[\hat{\theta}])^2 + E(\hat{\theta} - E[\hat{\theta}])^2\end{aligned}$$

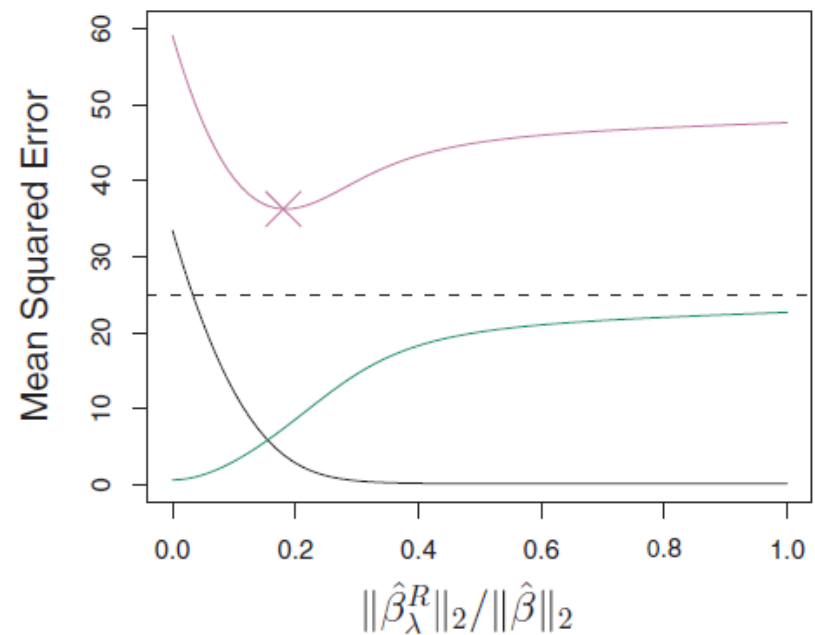
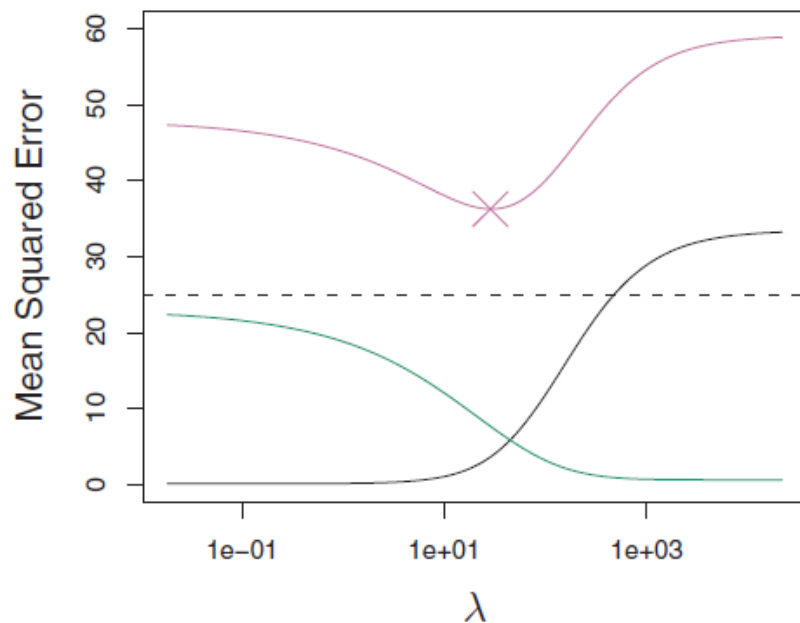
bias, squared   variance

- We can interpret the penalty term as a **bias**, pushing coefficients towards zero. The effect is a reduction on **variance**.



# Example: *Credit* data

- Squared bias in black, variance in green, MSE by cross-validation.



# **PENALISED REGRESSION WITH THE LASSO**

# Norms as Penalties

- The ridge regression penalty is known as the (square of)  **$l_2$  norm** (“ell 2”) penalty.

$$\sum_{i=1}^n \left( y^{(i)} - \beta_0 - \sum_{j=1}^p \beta_j x_j^{(i)} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

- Why do we use it? Among other things, it is differentiable. So, computationally convenient.
  - If you know what convex optimisation is, the above is also convex. The upshot? Unlike subset selection, all local minima agree on the same global minima!

# The $l_0$ penalty

- Subset selection can be seen as optimising this:

$$\sum_{i=1}^n \left( y^{(i)} - \beta_0 - \sum_{j=1}^p \beta_j x_j^{(i)} \right)^2 + \lambda \sum_{j=1}^p I(\beta_j \neq 0)$$

where  $I(z) = 1$  if  $z$  is true, 0 otherwise. This is the same as counting non-zero parameters.

- This is in one sense ideal, but computationally nasty, as we saw before. Definitely *not* differentiable, hence the combinatorial search.

# The Lasso

- Motivated as a “relaxation” of the  $l_0$  norm.

$$\sum_{i=1}^n \left( y^{(i)} - \beta_0 - \sum_{j=1}^p \beta_j x_j^{(i)} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

- Although not differentiable at zero, this is still a convex function and a variety of algorithms (some gradient-based) can be used to optimise it.

# Another Interpretation of Ridge Regression and the Lasso

Minimise

$$\sum_{i=1}^n \left( y^{(i)} - \beta_0 - \sum_{j=1}^p \beta_j x_j^{(i)} \right)^2$$

subject to

$$\sum_{j=1}^p |\beta_j| \leq s \quad (\text{Lasso})$$

or

$$\sum_{j=1}^p \beta_j^2 \leq s \quad (\text{Ridge})$$

where  $s$  is related to  $\lambda$ . You may recognize this from the idea of Lagrange multipliers.

# The Lasso's “Feasible Region”

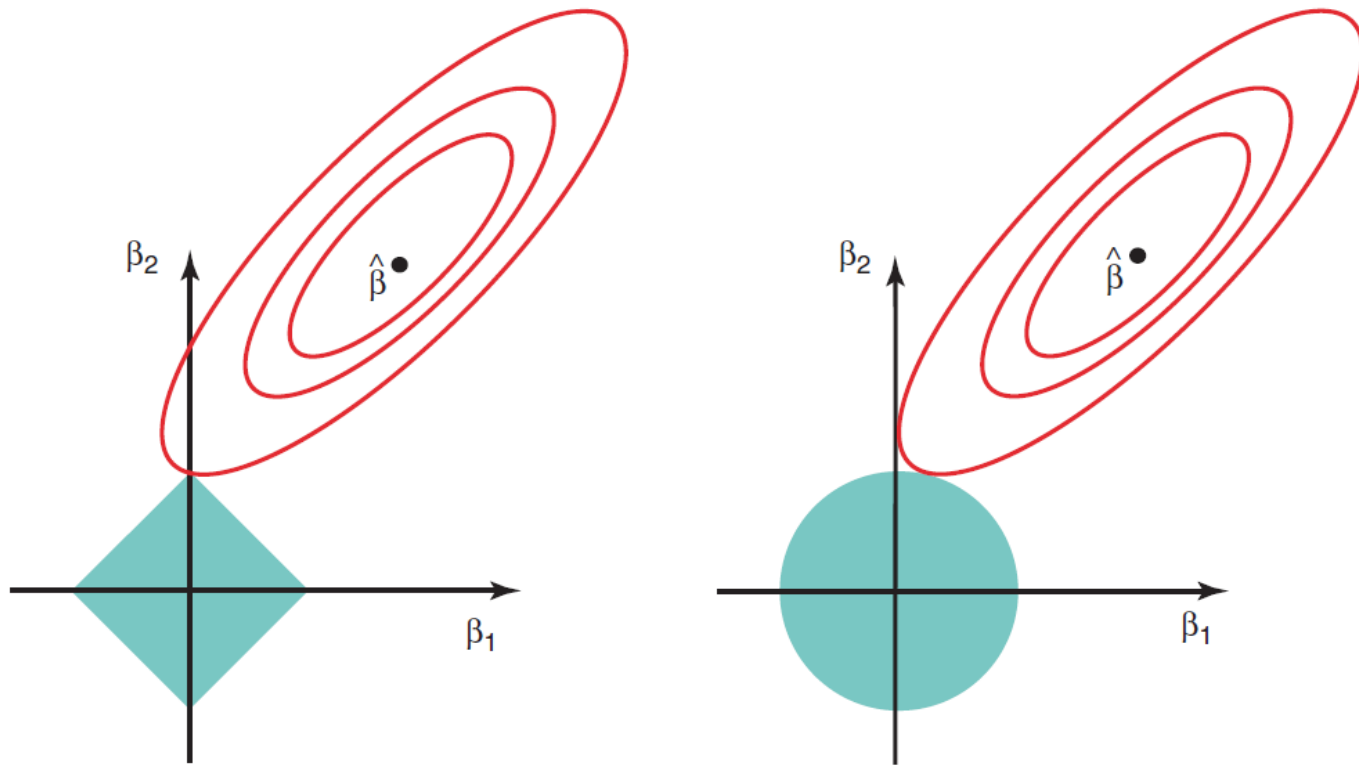
- Notice that the constraints for each coefficient in the lasso can also be written as:

$$-(s + \text{constant}) \leq \beta_j \leq s + \text{constant}$$

- if we fix the other coefficients.
- If we need to optimise a quadratic function within an interval, we can hit the boundary if the extreme of the quadratic function is outside of the feasible interval.

# The Effect of the $l_1$ Constraints

- This has a nice consequence: **sparsity**.



(red lines are the contour lines of the RSS, blue regions correspond to the penalisations)



# Overview: $l_0$ vs $l_1$ vs $l_2$

- $l_0$  can give sparse solutions, but it is not computationally tractable. Regularisation boils down to zero/one penalties.
- $l_2$  can give regularised solutions, but in general we will never see a sparse solution.
  - Informally we may think of thresholding, but this is not an optimal solution.
- $l_1$  can regularise and “sparsify” solutions in a single pass, being computationally convenient.
  - But this conflation of magnitude and sparsity is not ideal as they are not synonyms. Recall that a signal can be “large” and statistically non-significant, and “small” but significant.

# Notice

- We have only one penalty term (“**hyperparameter**”):

$$\sum_{i=1}^n \left( y^{(i)} - \beta_0 - \sum_{j=1}^p \beta_j x_j^{(i)} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

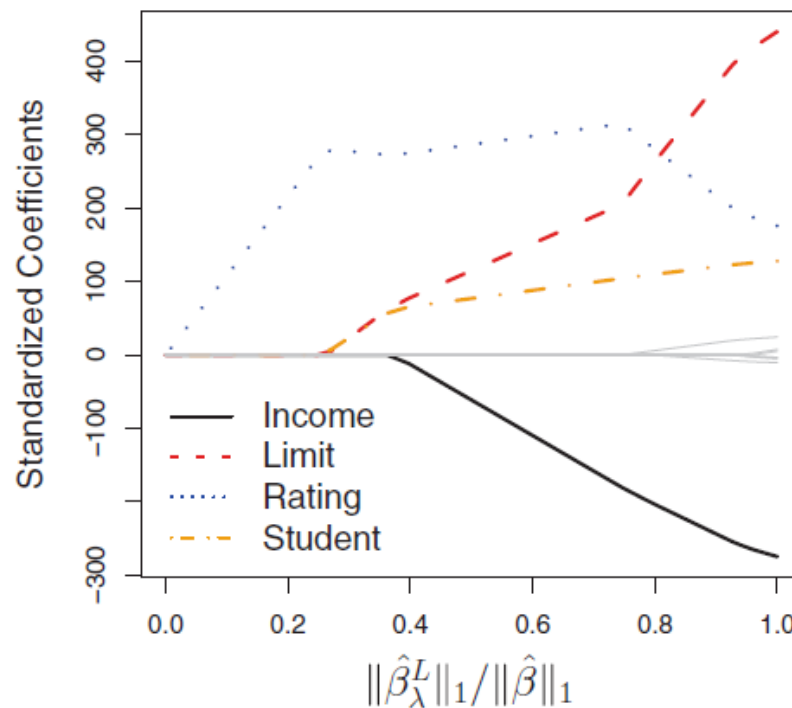
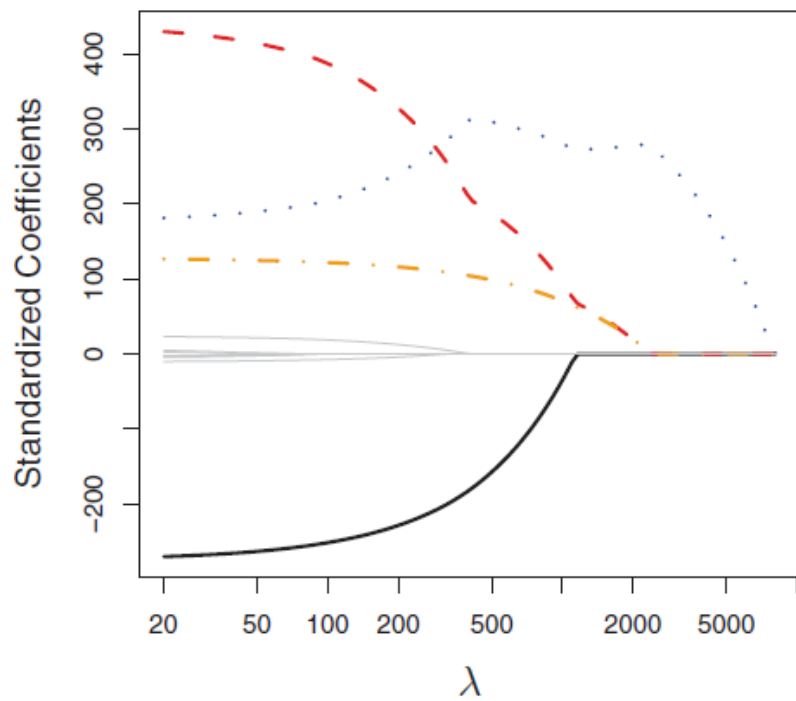
- This means that “large signals” (coefficients) will be “over-shrunk” if we push many other coefficients to zero.
  - There some (computationally expensive) Bayesian solutions based on using separate penalties per coefficient, and “regularizing the regularizers”.
  - In practice, lasso is much more widely used, and a non-Bayesian alternative of doing cross-validation across many penalty terms is hardly ever done as the number of combinations would explode easily.

# Notice

- Under some conditions,  $l_0$  and  $l_1$  agree on the same sparsity pattern.
- However, the usual conditions for that require that the dependency across the inputs is “weak”. So if you were able to solve  $l_0$ , we could expect to have a sparser solution.
- In particular, the problem would be “easy” if the inputs were all uncorrelated.
  - Exercise?

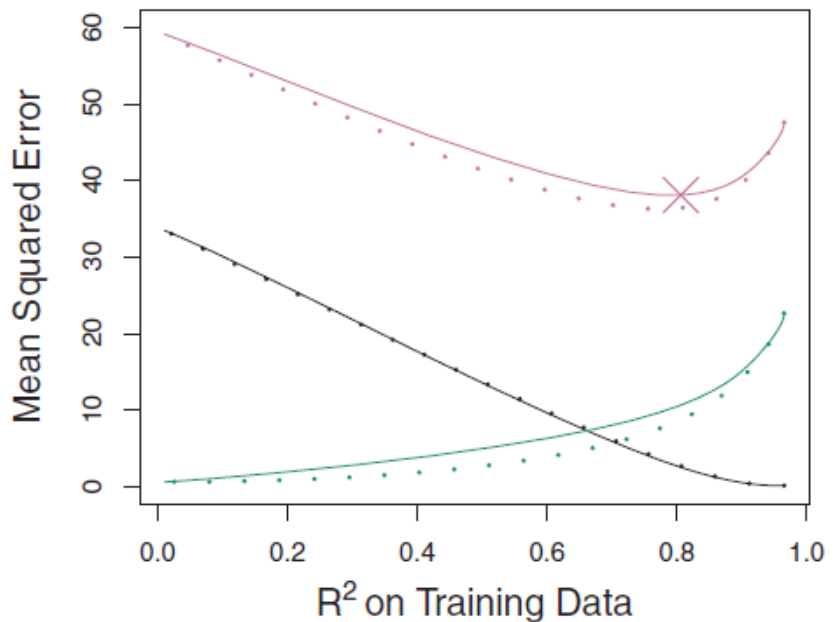
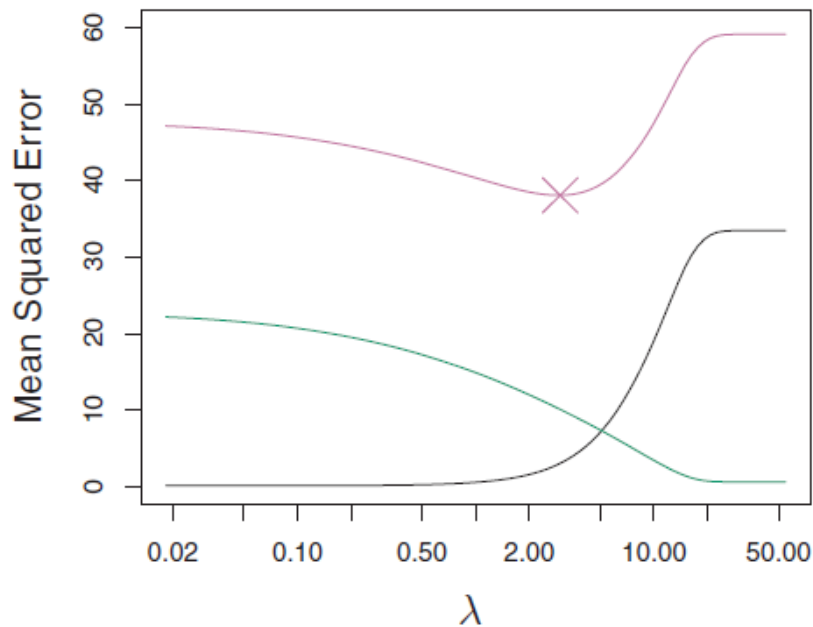
# Example: *Credit Data*

- Notice: standardization is used.



# Ridge vs Lasso

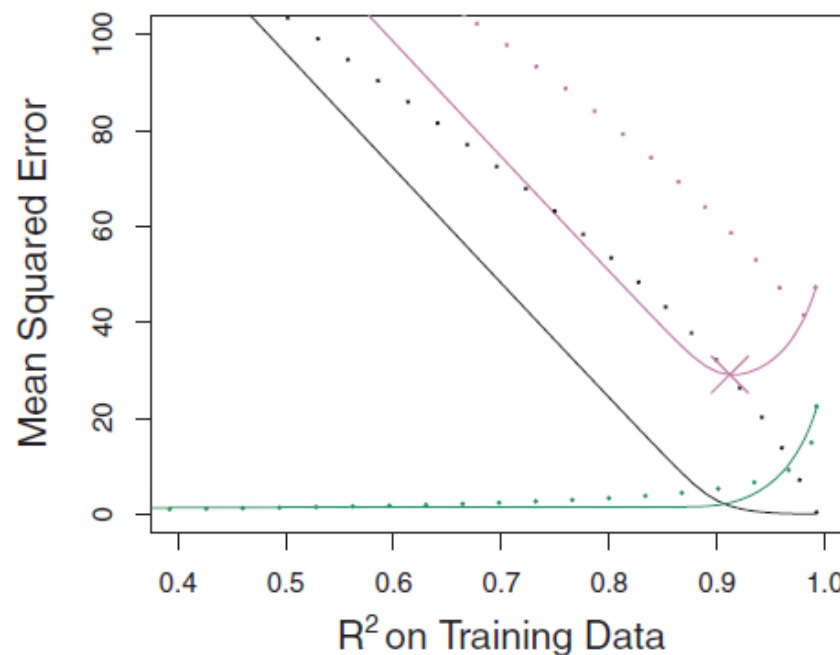
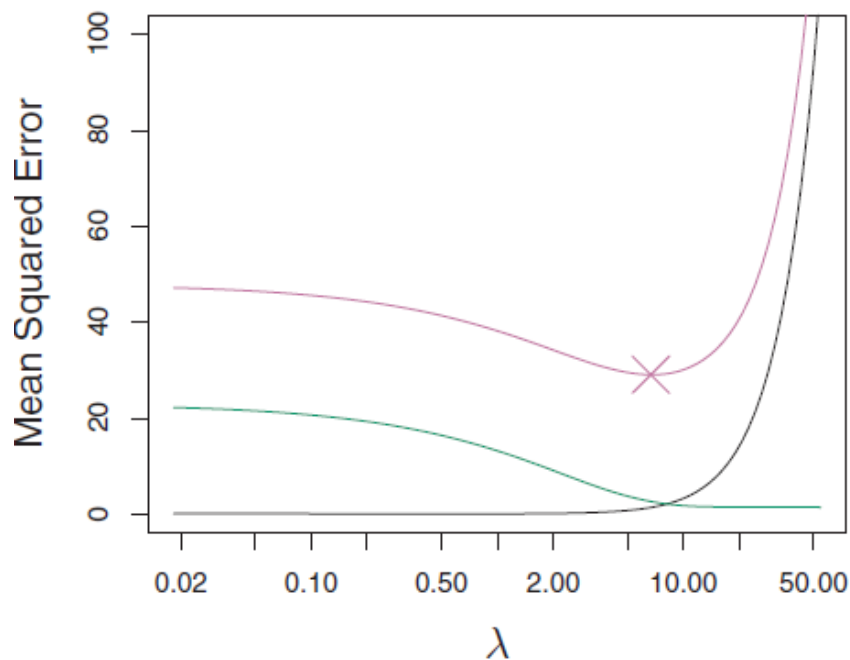
- MSE with simulated data. On the right, lasso is solid, ridge is dotted.



- Legend: squared bias (black), variance (green) & test MSE (pink).

# Notice

- In the previous simulation, no true sparsity existed. Let's see what happens when only 2 out of 45 variables contribute to the outcome.



# Walking Through a Simple Case

- We will go through a very simple artificial dataset to better highlight the differences.
- $n = p$ , and covariate training matrix  $\mathbf{X}$  is diagonal.
- Assume we will not have an intercept, to make things simpler. In least-squares, we get to minimise

$$\sum_{j=1}^p (y^{(j)} - \beta_j)^2$$

Solution?

# Solution

$$\hat{\beta}_j = y^{(j)}$$

What if we do ridge regression?

$$\sum_{j=1}^p (y^{(j)} - \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

We get

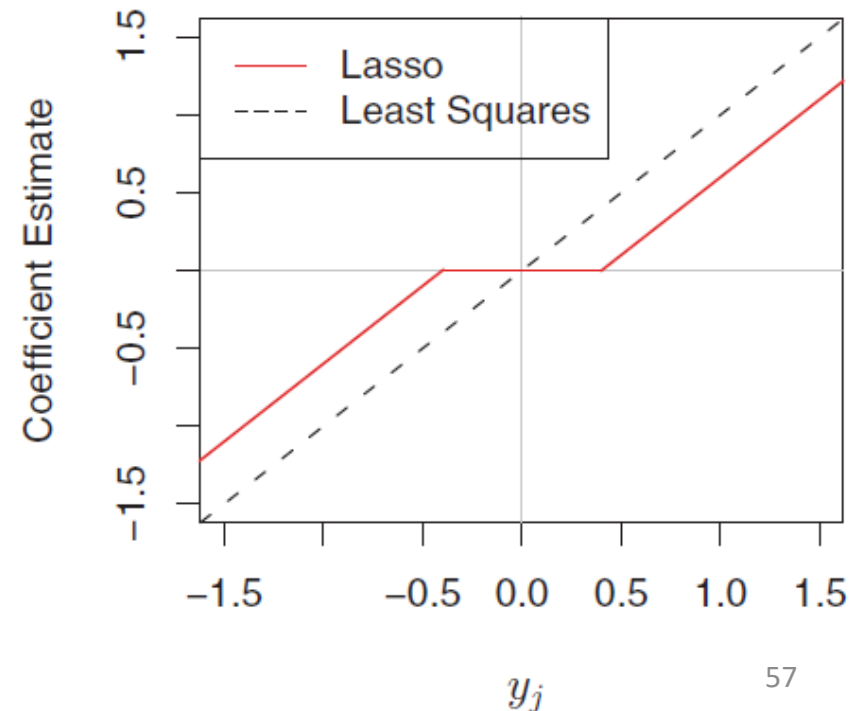
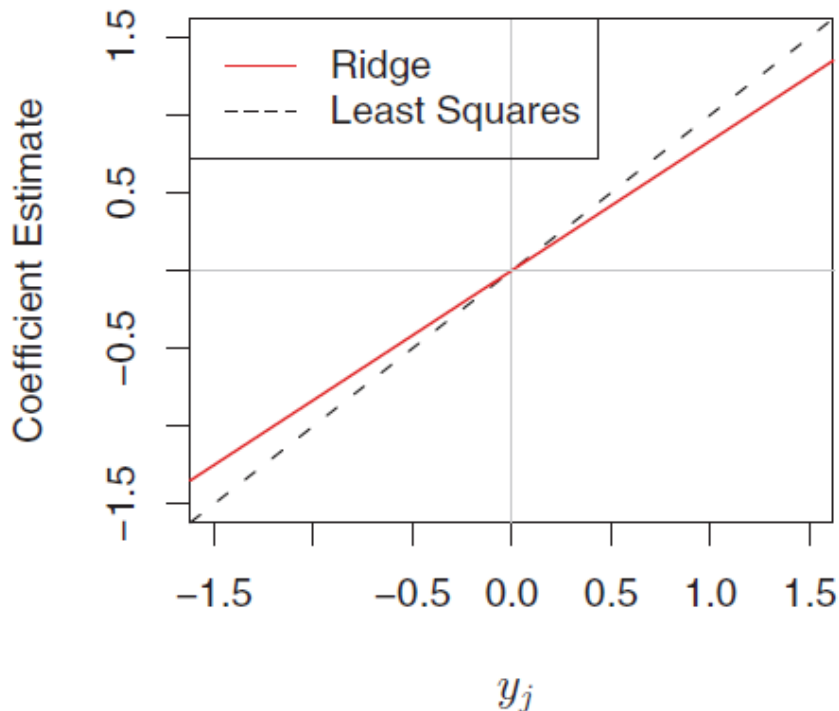
$$\hat{\beta}_j^R = y^{(j)} / (1 + \lambda)$$



# The Lasso Solution

■ That is,

$$\hat{\beta}_j^L = \begin{cases} y^{(j)} - \lambda/2, & \text{if } y^{(j)} > \lambda/2; \\ y^{(j)} + \lambda/2, & \text{if } y^{(j)} < -\lambda/2; \\ 0, & \text{if } |y^{(j)}| \leq \lambda/2; \end{cases}$$



# “Soft Thresholding”

- Both ridge regression and lasso shrink the least-squares estimates (push them towards zero).
- After a particular threshold, lasso shrink them all the way to zero.
- In real problems, lasso/ridge will perform better/worse depending how useful soft thresholding might be.
  - Lasso still better in terms of interpretability.
- It is possible to combine both penalties.

# A Note on Cross-Validation

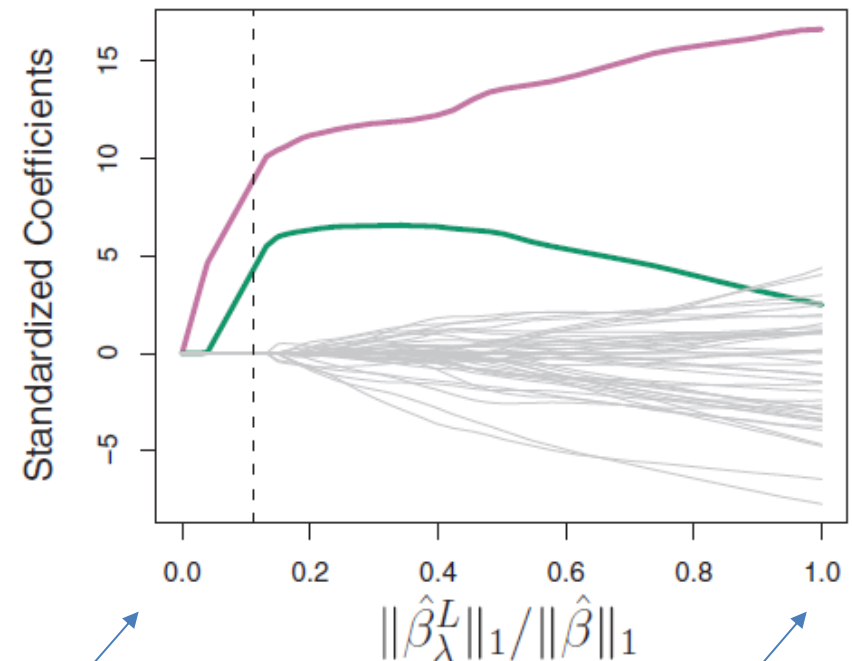
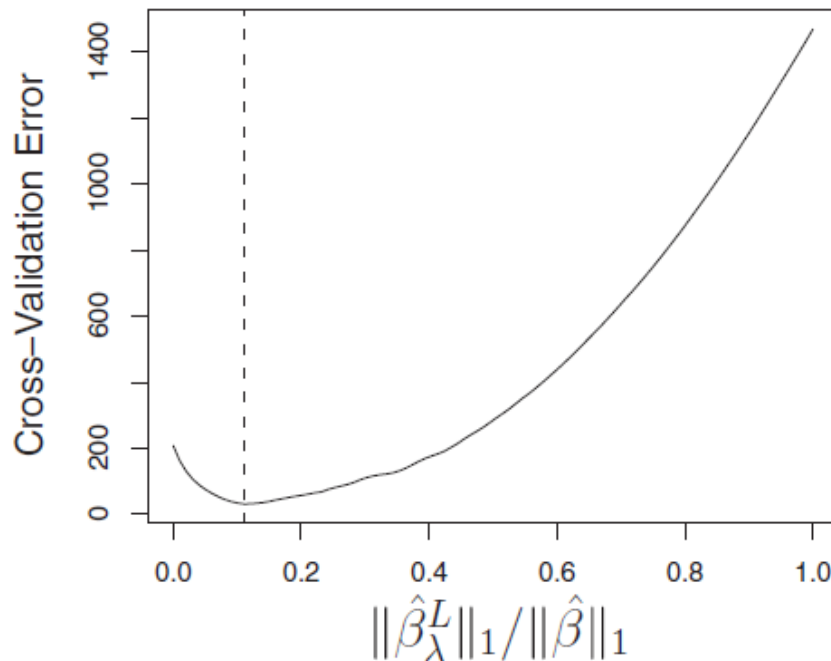
- If you implement cross-validation by yourself, you might be tempted to call a lasso optimiser repeatedly inside a loop over possible  $\lambda$ ..... Don't!
- There are algorithms for easily computing the “regularization path”, starting up lasso from the current value of  $\lambda$  to the next one in an efficient way.
  - R has great libraries for this, package **lars** is an example.
  - Also, manually setting up the range for  $\lambda$  may require some thought.

# Interpreting Cross-Validation Plots

- Be careful when interpreting regularization plots.
- You may see plots of what happens as training progresses (e.g. in neural networks software).
- Instead, you may also see what happens when a penalty parameter changes (e.g., the plots given by lars).

# Typical Regularization Path Plots

Simulation: 43 out of 45 variables irrelevant,  $n = 50$



Lasso + CV solution

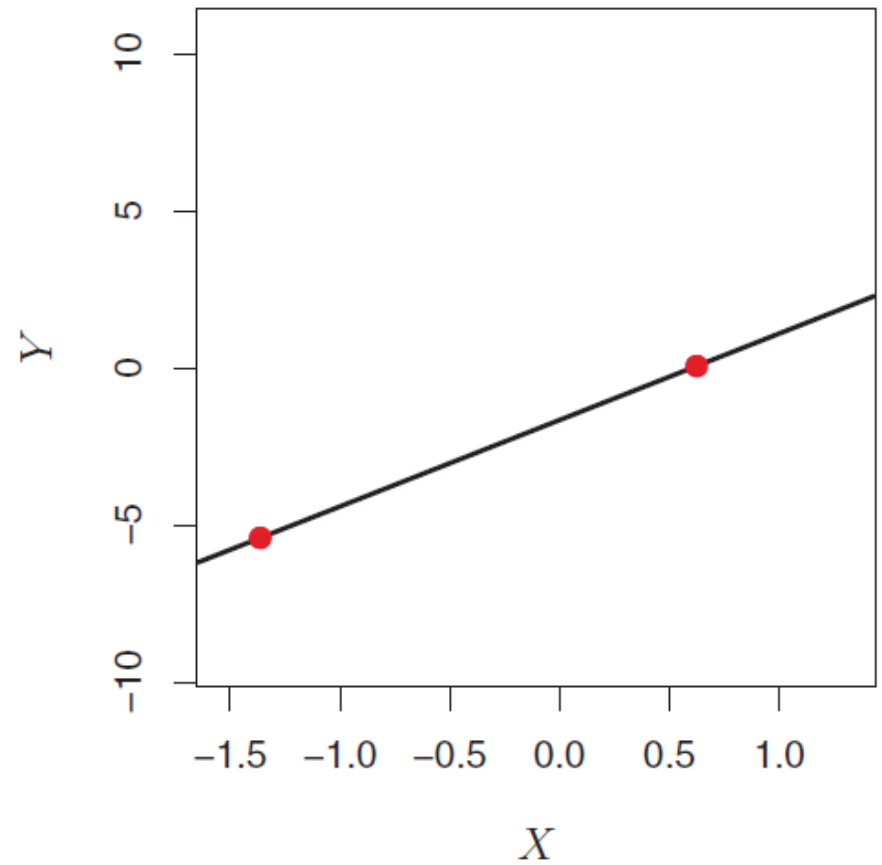
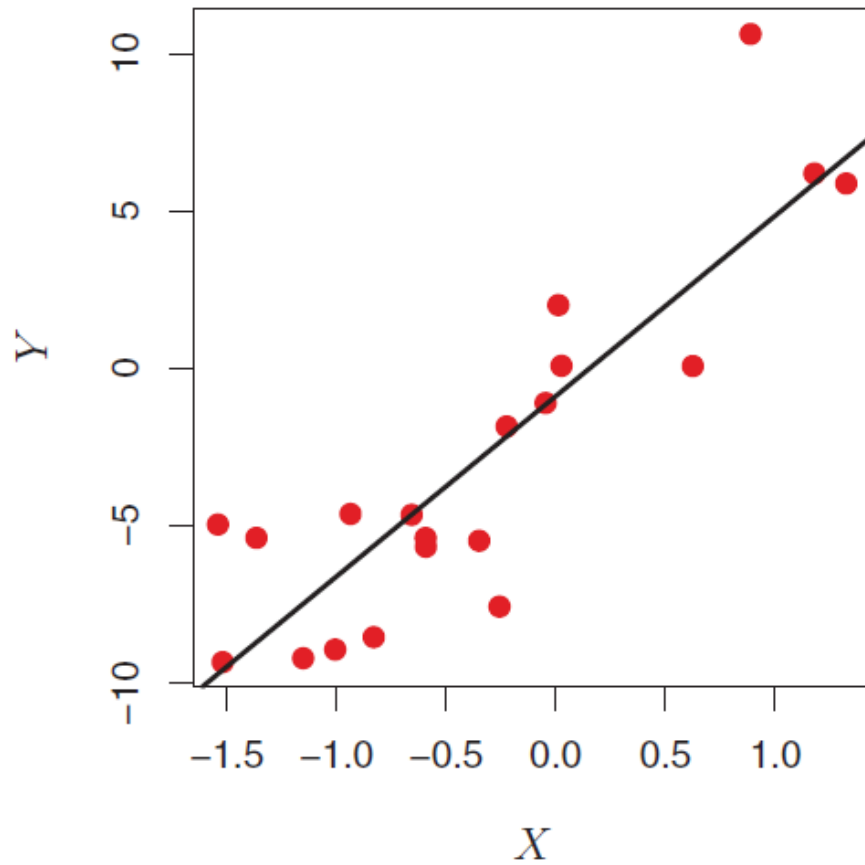
Least-squares solution!

# **HIGH-DIMENSIONAL REGRESSION**

# Technology as a Game Changer

- It is now possible to measure a large number of covariates.
- Examples:
  - high-throughput biological data, at the order of hundreds of thousands genetic features of a cell.
  - text/image data, assuming words/pixels as individual covariates.
- When data is abundant and interpretation is not relevant (e.g., second example above), methods such as modern neural networks might be a good go-to choice.
- However, when data is (relatively) scarce and interpretation is desirable, we need to rethink what regression means.

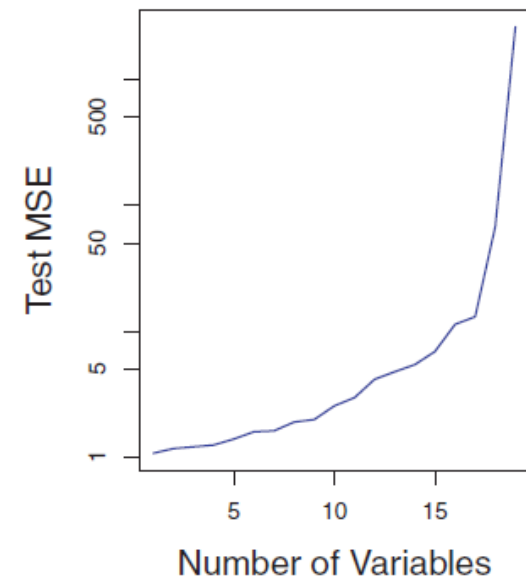
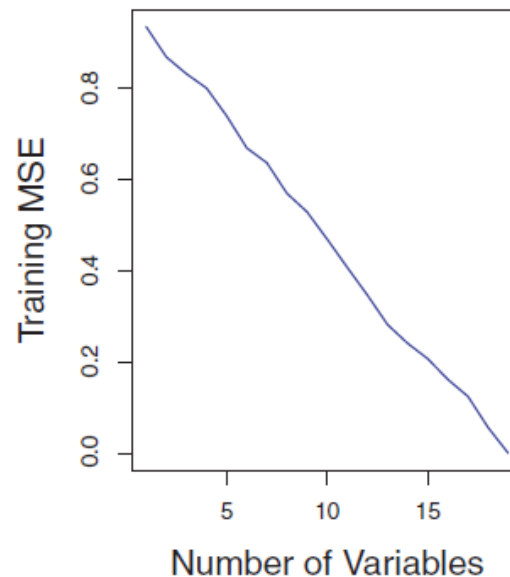
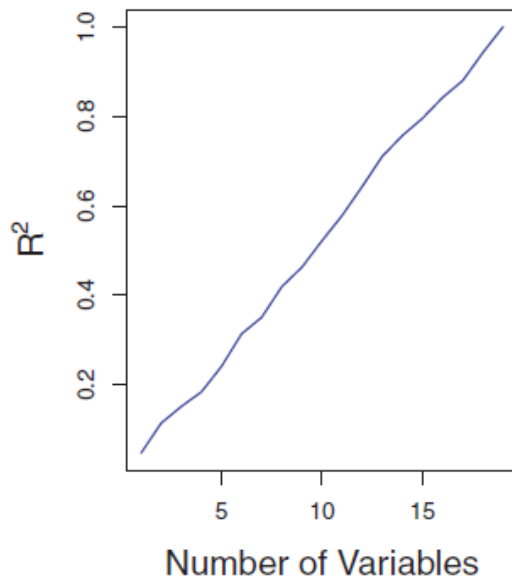
$$n = p$$





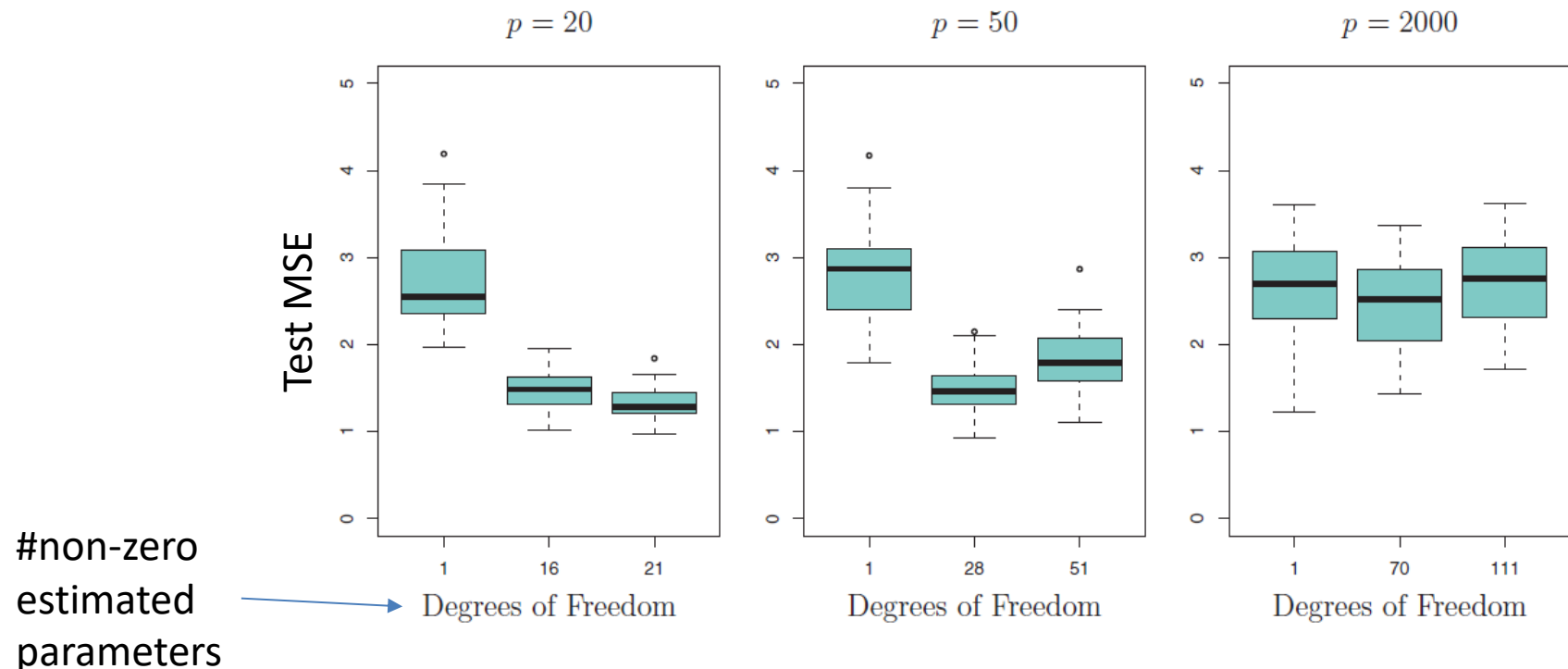
# $n = 20, p$ from 1 to 20

- All 20 covariates are unrelated to the outcome
- Likelihood depends on empirical variance, which is even more unstable than the mean:  $C_p$ /AIC/BIC useless as  $p$  increases.



# Simulation: Using Lasso, $n = 100$

- Only 20 features are related to the outcome.
- No cross-validation, just setting three levels of  $\lambda$ .



# Interpretation of High-Dimensional Regression

- It is hopeless to think we can learn a “true” model when  $p > n$  and there are many relevant covariates. This leaves us with two possibilities.
- First, learn a model “as good as it gets”.
  - We estimate a model that is the best in its class (linear model) and uses as many variables as it is feasible ( $n$ ).
  - More variables **is** more data after all. For prediction, it should make things **easier** not harder. We just need to make sure we don’t bite more than what we can chew on.

# Interpretation of High-Dimensional Regression

- Second, ***if*** the world is indeed sparse, that is, there are fewer than  $n$  relevant variables among those provided, ***then*** it is possible in theory to recover them.
- Statistically, some technical assumptions about
  - $n / p$  are necessary.
  - Computationally, some technical assumptions about the correlation of the inputs are necessary.
- This property is sometimes called “*sparsistency*”.

# In Practice

- What we recover is one of many possible models, one that can be statistically detected.
  - Training measures such as  $R^2$  will be useless here, but cross-validated predictions are meaningful.
- Prediction-wise, it is silly to throw variables away without looking at the data and without a theoretical justification, so we should welcome large  $p$ .
- Concerning what we learn about the world, we will be selecting promising explanatory variables as allowed by our data resources.

# Take-Home Messages

- Model search: two views
  - Finding real structure in nature (even if in practice it might be approximate)
  - Shrinkage/regularization: just minimise the effect of overfitting
- Computational considerations:  $l_0$ ,  $l_1$ ,  $l_2$  (and combinations) have their disadvantages and advantages.
  - Pragmatic advice: elastic net might be a good starting point.