Model Selection and Regularization

Mirko Draca

Warwick University

14th October 2020

Introduction

Why do we want to know this stuff?

- Gives us methods for dealing with the increasing number of high-dimensional data that is available. Key interesting examples include text data and genetic data.
- Introduces a further level of transparency on how we decide on variables to include in our models.
- Makes us think about the usefulness of predictive metrics. If a model predicts well but does not have an obvious behavioral / economic interpretation what is it telling us?

Statistical Learning and Economics.

- This is a set of ideas that have been developed in statistics and computer science and are now being integrated with economics and econometrics.
- One big challenge is the emphasis on prediction in statistical learning, whereas econometrics has a much bigger focus on inference issues.
- The other big challenge is finding applications that qualitatively add to what we have already been doing. Can these methods allow us to answer new questions that have been hard to crack following our usual research design formulas?

Some Reading

These are some papers that illustrate how model selection and regularization can be used in economics:

- Belloni,A; Chernozhukov,V; Hansen, C (2014) 'High-Dimensional Methods and Inference on Structural and Treatment Effects'. *The Journal of Economic Perspectives.* 28(2):29-50.

- Mueller-Smith, M (2015) 'The Criminal and Labor Market Impacts of Incarceration'. Mimeo, University of Michigan.

Some Reading

- Manresa, E (2016) 'Estimating the Structure of Social Interactions Using Panel Data'. Mimeo, NYU.

- Becker, S; Fetzer, T and Novy,D (2017) 'Who Voted for Brexit?' *Economic Policy* 32(92):601-650.

Finally, note that these notes draw heavily on the great Thiemo Fetzer's notes for the EC994 Data Science course. All credit for clarity goes to him!

THE PLAN

STATISTICAL LEARNING

BASIC LINEAR METHODS

SHRINKAGE METHODS

Statistical Learning

- Statistical learning refers to the set of methods to obtain robust predictive models.
- Suppose you have data on a set of variables $X_1, X_2, ..., X_p$, and we *assume* there exists a relationship between Y and $\mathbf{X} = (X_1, X_2, ..., X_p)$, that can be written as:

$$Y = f(\mathbf{X}) + \epsilon$$

- f represents the systematic way that $\mathbf{X}$ carries information about Y.
- Statistical learning refers to the different ways of estimating f.

Why would we want to Estimate f?

There are two main motivations to estimate f...

- **Prediction**: We want to forecast the value of Y based on X using $\hat{f}$ but don't necessarily care about getting 'true' estimates of β's in some population model.
- **Inference**: Which variables $X_1, X_2, ..., X_p$ are associated with Y?
  - Are the effects meaningful in terms of size? Do effects work through the interaction?
  - Is linearity an adequate assumption?

How to Assess Model Accuracy? In Theory...

- We need a way to measure and evaluate the relative performance of different statistical learning methods. In regression framework, the most commonly used objective is "mean squared error" (MSE), defined as:

$$MSE = 1/n \sum_{i=1}^{n} \hat{e}_i{}^2 = 1/n \sum_{i=1}^{n} (\hat{f}(x_i) - y_i)^2$$

- Assuming f, X are constant, MSE is an estimate of

$$\begin{aligned} E(Y - \hat{Y})^2 &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\ &= [f(X) - \hat{f}(X)]^2 + Var(\epsilon) \end{aligned}$$

- While the first part is "reducible" error that can be removed by improving the way we estimate the true f, the second component $Var(\epsilon)$ is "irreducible".

In practice, we operationalize the MSE concept by applying a **'validation set'** approach on our data....

Validation Set Approach to Assess Model Accuracy

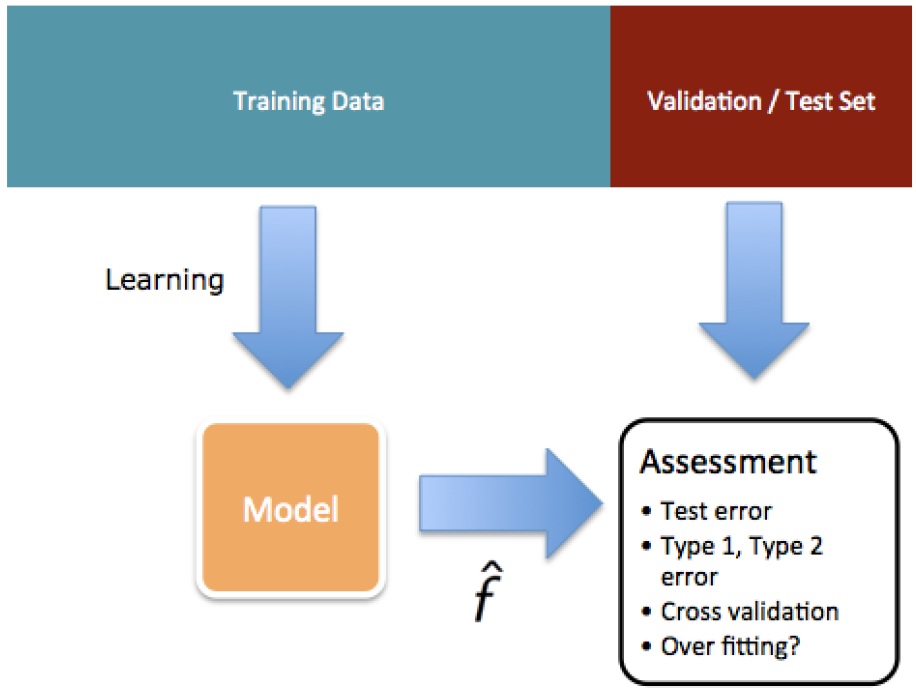

Figure 1: Validation Set Workflow Illustration.

Validation Set Approach to Assess Model Accuracy

- $\hat{f}$ is estimated using only a subset of the data, known the "training set". This is the part of the data that was used to estimate f. We compute the **training MSE**.
- On the rest of the data, the *test data*, we study how well the estimated $\hat{f}$ performs on virgin data that has not been used to fit the model. You can think of the test data as new data for which you want to *predict* the outcome.

  Based on this, we compute the **test MSE** and we would like this test MSE to be as small as possible: the smaller **test MSE**, the better is our predictive model.
- Its important that the training and test data are randomly selected subset of the data; otherwise, you may introduce systematic differences between the training dataset and the test dataset.

Overfitting...why does this happen?

- Fitting ever more complicated models (eg:loads of polynomials), while ignoring the true model implies, that the estimated models are starting to **explain the noise** contained in ϵ.
- From Econometrics: including *irrelevant variables* (that is those with coefficients ≈ 0) does not result in biased point estimates, but the resulting estimators are not *efficient*, i.e. OLS is not BLUE.
- Since the noise is randomly drawn and thus, the noise in the training set is **independent** from the noise in the test set, the performance of the fit estimated from the data in the training set will become worse and worse

Overfitting...why does this happen?

- Its important that you do not look at the test set, as otherwise this independence assumption may be violated, if you eg. adjust the set model to capture some features in the training set.
- In Applied Economics, people generally only look at models explaining a given data set (the training data), but do not assess model performance out of sample.
- In reality, we do not know what the true function for f is...hence, in order to assess whether we are overfitting the data, we need to study test MSE as we try different methods of estimating f.

Bias vs Variance Tradeoff

- We can more formally describe what is happening to the test MSE as model complexity increases.
- The U-shape of the test MSE is driven by two competing forces
- The expected test MSE at any different point x_0 can be decomposed as:

$$E(f(x_0) - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

where $Bias(\hat{f}(x_0)) = E(f(x_0) - \hat{f}(x_0))$. Can you show this?

Bias vs Variance Tradeoff Proof [advanced]

Show that:

$$E(f(x_0) - f(\hat{x_0}))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

Proof:

- To get rid of the indices, let $f = f(x_0)$, and $\hat{f} = \hat{f}(x_0)$.
- Remember that for some random variable X, $Var(X) = E[(X - E(X))^2] = E(X)^2 - E(X^2)$, so in particular $Var(\epsilon) = E(\epsilon^2)$, since $E[\epsilon] = 0$.
- Further: $E(y) = f$, since $y = f + \epsilon$ and $E(\epsilon) = 0$ and f is itself not a random variable.
- This implies $Var(y) = E[(y - E(y))^2] = E[(f + \epsilon - f)^2] = E[\epsilon^2] = Var(\epsilon)$
- Also remember that for some random variables X, Y: $Cov(x, Y) = E(XY) - E(X) * E(Y)$
- So in particular: $E(\hat{f}\epsilon) = 0$, since ϵ from test set is randomly and independently drawn from prediction $\hat{f}$.

Together this yields

$$\begin{aligned}
\mathrm{E}\big[(y - \hat{f})^2\big] &= \mathrm{E}[y^2 + \hat{f}^2 - 2y\hat{f}] \\
&= \mathrm{E}[y^2] + \mathrm{E}[\hat{f}^2] - \mathrm{E}[2y\hat{f}] \\
&= \mathrm{Var}[y] + \mathrm{E}[y]^2 + \mathrm{Var}[\hat{f}] + \mathrm{E}[\hat{f}]^2 - 2f\mathrm{E}[\hat{f}] \\
&= \mathrm{Var}[y] + \mathrm{Var}[\hat{f}] + (f - \mathrm{E}[\hat{f}])^2 \\
&= \mathrm{Var}[\epsilon] + \mathrm{Var}[\hat{f}] + [\mathrm{E}(f - \hat{f})]^2
\end{aligned}$$

Bias vs Variance Intuition

$$E(f(x_0) - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

Look at the individual elements here:

$Var(\epsilon)$

... is a constant.

it remains unchanged for different $\hat{f}$'s.

it represents the lowest bound for a test error that is attainable, since both the other terms are positive.

Minimizing test error requires finding an $\hat{f}$ that minimizes the sum of squared bias and variance.

Bias vs Variance Intuition

$$E(f(x_0) - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

Look at the individual elements here:

$Var(\hat{f}(x_0))$

... refers to the amount by which $\hat{f}$ would change if we estimated it *using a different training data set.*

Since the training data are used to fit the statistical learning method, different training data sets produce different $\hat{f}$. Ideally the estimate for f should not vary too much between training sets.

Different methods have different variances: more flexible methods have larger variances, while less flexible ones (e.g. linear regression) have lower variance.

This is pushing up our test MSE for highly flexible specifications.

Bias vs Variance Intuition

$$E(f(x_0) - \hat{f(x_0)})^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

Look at the individual elements here:
$[Bias(\hat{f}(x_0))]^2$

... we're working with an approximate model, that leaves out relevant factors systematically introduces errors by not allowing for more complex interactions between variables X_i.

e.g. a linear model may be inadequate in case the true relationship is non-linear, introducing significant bias.

This is akin to the idea of ommitted variable bias in regression, which causes the true effect of some variable to be under or over-stated, thus, distorting the predictive power of that variable.

BASIC LINEAR METHODS + MODEL SELECTION CRITERIA

Linear Regression Revisited

- The linear regression model gas the form:

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \tag{1}$$

- Assumption: Regression function $E(Y|X)$ is linear or approximately linear, where β_j are to be estimated.
- Typically the X_j's come from different sources:
  - quantitative inputs and transformations of them (eg. log)
  - polynomial base expansions, e.g. for two $p = 2$, a second order expansion would imply a total of five regressors $X_1, X_1^2, X_1 * X_2, X_2^2, X_2$
  - Numeric categorical variables (e.g. points on a Likert scale).
- Assume X has dimensions $N \times p$, i.e. N rows and p columns (regressors).

Least squares fit

- Least square fit solves the following optimization problem

$$argmin_{\beta} RSS(\beta) = (y - \mathbf{X}\beta)'(y - \mathbf{X}\beta) \tag{2}$$

- This is a matrix way of writing: find the vector $\beta = (\beta_1, ..., \beta_p)$, such that:

$$\sum_{i=1}^{n} (y_i - \sum_{k} x_k \beta_k)^2$$

  is minimized.
- Solution obtain by solving FOC:

$$\frac{\partial RSS}{\partial \beta} = -2\mathbf{X}'(y - X\beta)$$

- and setting equal to zero

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

Some properties of the fit

- Least squares solution is the best linear unbiased estimator if the y's are conditionally indepedent for a given set of inputs x_i (Gauss Markov Theorem)
- In case of homoskedastic errors, $Var(\hat{\beta}) = (X'X)^{-1}\sigma^2$
- The σ^2 is typically estimated by

$$\hat{\sigma}^2 = \frac{1}{N - p - 1} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

- It can be shown that $E(\hat{\sigma^2}) = \sigma^2$, i.e. $\hat{\sigma^2}$ is an unbiased estimator of the population variance.
- With normally distributed error term, i.e. $\epsilon \sim N(0, \sigma^2)$, one can show that

$$\hat{\beta} \sim N(\beta, (X'X)^{-1}\sigma^2)$$

- The distribution of β can be estimated, when plugging in an estimate of σ^2. This can be used for hypothesis testing on the effects of β_i's.

Further properties of the fit

- We can express the variability in y as

$$TSS = \sum (y_i - \bar{y})^2$$

- Of which a regression explains a proportion

$$ESS = \sum (\hat{y}_i - \bar{y})^2$$

- And leaves unexplain

$$RSS = \sum (y_i - \hat{y}_i)^2$$

- $TSS = ESS + RSS$ [Do you remember how to show this?]
- A measure of goodness of fit is given as

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

- Its hard to say, what is a "good" R^2 - why? *RSS* only knows one direction... it goes down as we include more controls, even if these controls dont have any empirical content. They simply fit the noise in the error term ϵ.

Some Important Questions

1. Is at least one of the predictors $X_1, ..., X_n$ useful in predicting the response?
2. Do all predictors help explain Y, or is only a subset useful?
3. How well do we fit the data? How much is left unexplained?
4. How well do we predict?

Extra Practical Questions

- What if $p > N$ or approaches N? How can we choose what X_p features to include in a $f(x)$?
- Do some methods have advantages when we think many p are non-trivially important?
- How can we explore the space of the potential $f(x)$ at a lower computational cost?

Tractor Methods

I call these 'tractor' methods because they involve running lots of lots of combinations of X_p iteratively and then looking for the best fit.

- Best Subset Selection.
- Forward Stepwise Selection.
- Backward Stepwise Selection.

...note that I am the only person in the world (that I know of) who calls this a 'tractor' approach!

Best Subset in a Nutshell

- We are basically 'trying out' different numbers of p to find the best combination of X_p's at each $p = k$.
- That is, we find the best 2-feature model out of all possible p, then the best 3-feature model, and so on. These are the 'best subsets' for each $p = k$ level.
- We then compare across the different $p = k$ levels by using some type of penalized model fit statistic.

Subset Selection

Suppose you have p predictors, that is $X_1, ..., X_p$ and you want to identify the best subset of predictors M, possibly with $M < p$, that achieve "best performance".
This is a difficult task...why?

- There are p models, containing exactly 1 predictor
- There are $\binom{p}{2} = p(p-1)/2$ possible ways to choose 2 predictors [Remember: $\binom{p}{k} = \frac{p!}{k!(p-k)!}$]
- There are $\binom{p}{3} = \frac{p!}{3!(p-3)!} = p(p-1)(p-2)/6$
- ...

In total there are: $\sum_{k=1}^{p} \binom{p}{k} = 2^p$ possible models. How would we proceed to find the "best" possible model?

Best Subset Algorithm

1. Let $\mathcal{M}_0$ denote the null model containing no predictors except for a constant (i.e. predicting the mean).
2. For $k = 1, ..., p$

   a) Fit all $\binom{p}{k}$ models that contain k predictors.

   b) Pick the best among these k dimensional models, calling it $\mathcal{M}_k$. The *best* model is the one that has smallest RSS or largest R^2.
3. Select the single best model from the set $\mathcal{M}_0, ..., \mathcal{M}_p$. Here *best* is determined using best performance in terms of MSE on a training set, or some measure of goodness of fit that adjusts for the fact that R^2 monotonically decreases as k gets larger.

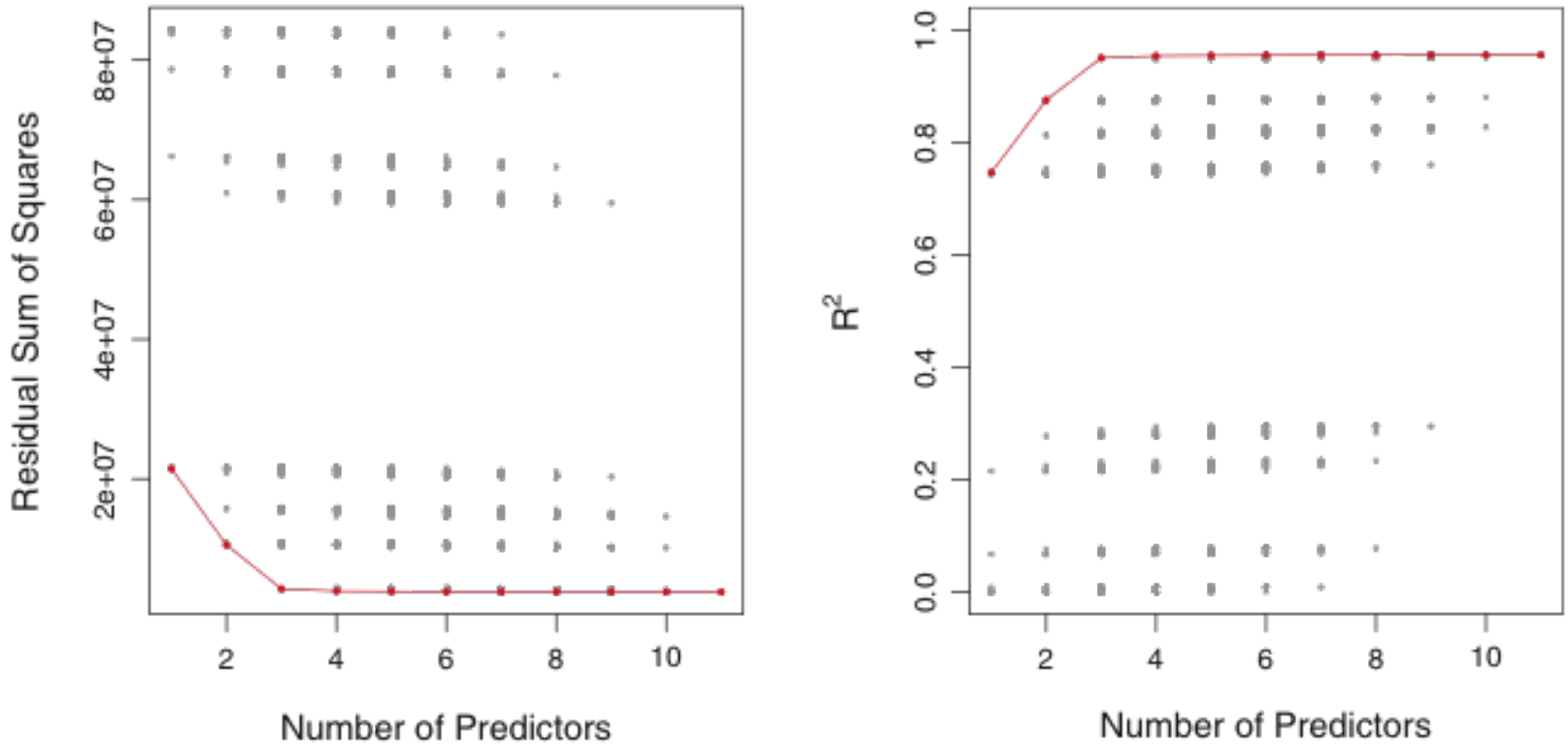

FIGURE 6.1. *For each possible model containing a subset of the ten predictors in the* `Credit` *data set, the RSS and* R^2 *are displayed. The red frontier tracks the* best *model for a given number of predictors, according to RSS and* R^2*. Though the data set contains only ten predictors, the* x*-axis ranges from* 1 *to* 11*, since one of the variables is categorical and takes on three values, leading to the creation of two dummy variables.*

Some Points to Note...

- You know that R^2 monotonically increases as you add more control variables. Since you hold k fixed, i.e. you compare only the performance of models with the same set of predictors, you can choose the one with the largest value of R^2, since you compare "like with like".

- Here, we are not comparing "like with like", but rather we want to penalize models that are mechanically bound to perform better in the training sample; we can do this by computing their performance in terms of MSE on a training set, or, in absence of a training set, we can look at various statistics that adjust the measure of goodness of fit.

Some Points to Note...

- The computational cost of Best Subset is huge. eg: If $p = 10$ then there are $2^p = 1,024$ models and for $p = 20$ a bit over 1 million.
- There are some formal shortcuts (eg:'branch and bound') plus we can do *ad hoc* pruning based on what we think is relevant to the application.
- However, some simple modifications exist in the form of stepwise selection...

Forward Stepwise Selection in a Nutshell.

- This is a less intensive alternative to Best Subset
- We start with a null ($p = k = 0$) model and then add features in a stepwise fashion, choosing the 'best' one at each iteration..
- Then we choose across the 'best at step k' models using some model selection criteria.

Forward Stepwise Selection Algorithm

1. Let $\mathcal{M}_0$ denote the null model containing no predictors except for a constant (i.e. predicting the mean).
2. For $k = 0, ..., p-1$

   a) Consider all $p-k$ models that augment the predictors of $\mathcal{M}_k$ with one additional predictor.

   b) Choose the best among these $p-k$ models and call it M_{k+1}. The *best* model is the one that has smallest RSS or largest R^2.
3. Select the single best model from the set $\mathcal{M}_0, ..., \mathcal{M}_p$. Here *best* is determined using best performance in terms of MSE on a training set, or some measure of goodness of fit that adjusts for the fact that R^2 monotonically decreases as k gets larger.

Computational Demand of Forward Stepwise...

Where or how do we save time here?

- Each forward step involves fitting only $p - k$ models in each iteration k, rathar than $\binom{p}{k}$.
- So in first iteration, we fit p models, in second, we fit $p - 1$, in third ...
- This means, in total we fit $\sum_{k=0}^{p-1}(p-k) = 1 + p(p+1)/2$ models.
- In the case of $p = 40$, this amounts to 821 models.

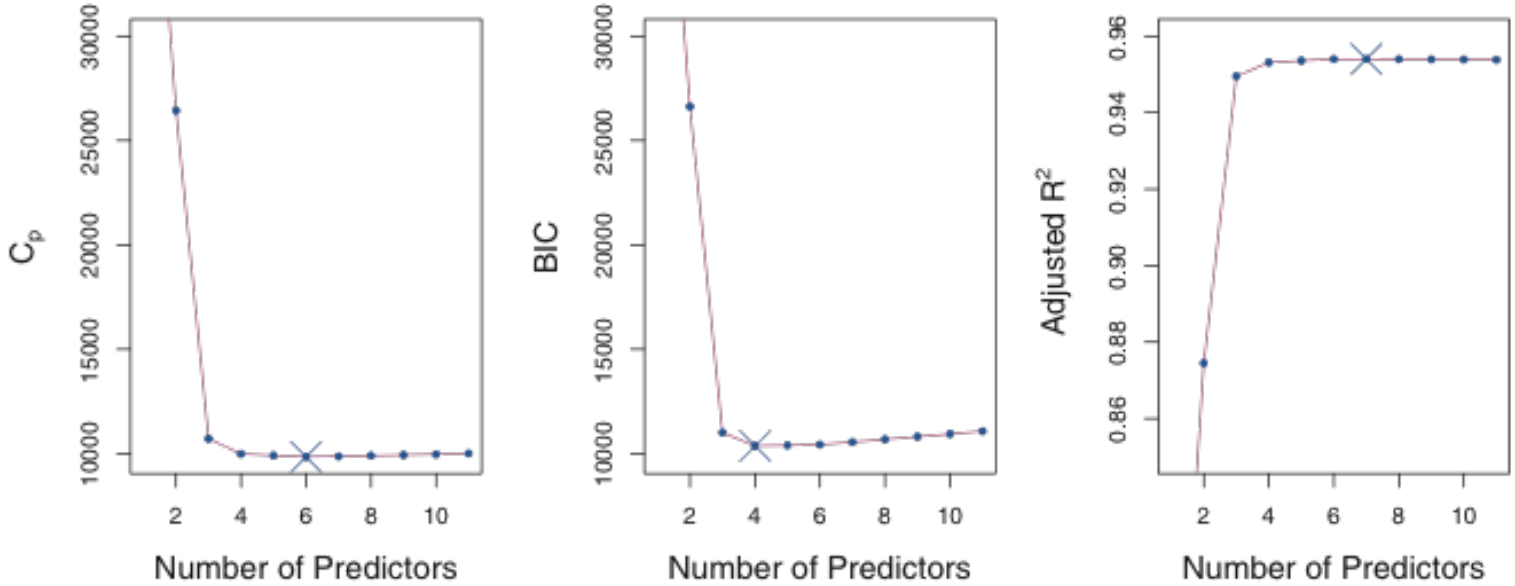

FIGURE 6.2. *C_p, BIC, and adjusted R^2 are shown for the best models of each size for the* `Credit` *data set (the lower frontier in Figure 6.1). C_p and BIC are estimates of test MSE. In the middle plot we see that the BIC estimate of test error shows an increase after four variables are selected. The other two plots are rather flat after four variables are included.*

# Variables	Best subset	Forward stepwise
One	`rating`	`rating`
Two	`rating`, `income`	`rating`, `income`
Three	`rating`, `income`, `student`	`rating`, `income`, `student`
Four	`cards`, `income`, `student`, `limit`	`rating`, `income`, `student`, `limit`

TABLE 6.1. *The first four selected models for best subset selection and forward stepwise selection on the* `Credit` *data set. The first three models are identical but the fourth models differ.*

Backward Stepwise Selection in a Nutshell.

- This is the inverse of the forward method: it starts with including all the p variables and then drops variables one by one.
- It cannot be used so well when $p > N$ since we run out of degrees of freedom.
- The sleeper issue with these methods is collinearity across the p variables. The importance of one particular p will often hinge on what other p is in the model.

Backward Stepwise Selection Algorithm

1. Let $\mathcal{M}_p$ denote the *full* model containing all p predictors.
2. For $k = p, p-1, ..., 1$

   a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, i.e. that contain a total of $k-1$ predictors.

   b) Choose the best among these k models and call it M_{k-1}. The *best* model is the one that has smallest RSS or largest R^2.
3. Select the single best model from the set $\mathcal{M}_0, ..., \mathcal{M}_p$. Here *best* is determined using best performance in terms of MSE on a training set, or some measure of goodness of fit that adjusts for the fact that R^2 monotonically increases in more complex models.

Lets pick the "best" $\mathcal{M}_k$ - Model Selection.

As discussed, R^2 is adequate to choose the best model, when we compare "like with like", i.e. models of the same number of parameters; however, we can not compare them across different values of k. We want to select the model with the lowest test error, i.e. the model that provides most robust predictions and does not suffer from "overfitting". There are two approaches...

1. We can add a penalty term to a measure of goodness of fit for the training set, to account for the bias that is likely to arise due to overfitting. The statistics that do this are AIC and adjusted R^2.
2. Directly estimate the test error, by computing *test MSE* for the different models based on a validation set.

Approach (1) is usually followed, in case no test data set is available.

AIC and Adjusted R^2

- From earlier, we know that the training set MSE is an underestimate of the of the test MSE [Remember $MSE = RSS/n$]
- That is, the MSE estimated from the training set is *too low* relative to the true test MSE.
- We can correct for this "optimism" directly.
- The main one we discuss is the *Akaike information criterion* for a model with k predictors.

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2k\hat{\sigma}^2)$$

- It's beyond the interest level of this course to derive where this is coming from... but, it is built up in an MLE framework.
- $\hat{\sigma}^2$ is an estimate of the variance of the residuals ϵ.
- As we increase k, the second term becomes larger. The model fit is better, **the smaller** AIC.

AIC and Adjusted R^2

$$Adjusted R^2 = 1 - \frac{RSS/(n-k-1)}{TSS/(n-1)}$$

- Note that the denominator is constant. Maximizing Adjusted R^2 is equivalent to minimizing $\frac{RSS}{(n-k-1)}$
- However, the numerator changes non-linearily in k.
- As $k \uparrow$, $RSS \downarrow$ and $(n-k-1) \downarrow$.
- So R^2 keeps increasing, if the decrease in RSS is larger than the decrease in $(n-k-1)$.

Intuition: Once all the "correct" variables have been included in the model, additional *noise* variables will lead to only a small decrease in RSS, while the denominator $(n-k-1)$ decreases a lot, so the ratio increases and adjusted R^2 goes down.

Model Selection Criteria (MSC)

- Other MSC include the C_p statistic and the Bayesian Information Criteria (BIC). You can read about them in *Introduction to Statistical Learning* Section 6.1.3.
- One major point about adjusted R^2 is that it lacks a rigorous theoretical justification in the same way as these other MSC.
- Aside: I think what's missing in the 'Data Science' approach is judgments around collinearity and the generating model. As researchers we might think that there are natural subsets of variables (eg: soft skills, hard skills) and we can use this to discipline p selection. But this has to be transparent...

‘SHRINKAGE’ METHODS –

RIDGE REGRESSION AND LASSO.

Introduction to Shrinkage Methods

- You can summarize Subset methods as finding the 'minimal' models in terms of p that robustly predict out of sample. Practically, this involves trying out lots and lots of models.
- In contrast, shrinkage methods set up an objective function that considers the contribution of the different p in what is effectively a single 'step'.
- This step can be thought of as *constraining* or *regularizing* the coefficients and, practically, *shrinks* the estimates to zero.

Setting up Ridge Regression (RR)

The RR approach is set up in terms of this objective function:

$$argmin_{\beta} \underbrace{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_k \beta_k)^2}_{\text{Residual Sum of Squares}} + \underbrace{\lambda \sum_{j=1}^{p} \beta_j^2}_{PenaltyTerm}$$

Hence it boils down to the the part we know from OLS (the RSS) plus a constraint framed in terms of the (summed) value of the β coefficients.

The λ term is a *tuning parameter* that determines how heavy the penalty is and note that we use a simple square or quadratic function to make the β's non-zero in terms of the objective function.

Ridge regression [a bit mathsy]

$$argmin_\beta \underbrace{\sum_{i=1}^{n}(y_i - \sum_j x_{ij}\beta_j)^2}_{\text{Residual Sum of Squares}} + \underbrace{\lambda \sum_{j=1}^{p}\beta_j^2}_{PenaltyTerm}$$

We can rewrite the objective function in matrix notation as:

$$argmin_\beta (y - \mathbf{X}\beta)'(y - \mathbf{X}\beta) + \lambda\beta'\beta$$

- This is very close to the objective function for plain OLS.
- The First Order Conditions are:

$$-2\mathbf{X}'(y - X\beta) + 2\lambda\beta$$

- Setting equal to zero

$$\mathbf{X}'y = (\lambda I + \mathbf{X}'X)\beta$$

- From this follows directly:

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

Ridge regression

- You can see directly that if $\lambda = 0$, then the solution is just OLS.
- As $\lambda \to \infty$, $\beta \to 0$.
- For every value of λ you get a different estimated coefficient vector.
- Note that we don't have a penalty for the intercept β_0, which provides the mean value of the dependent variable y, when all the x_i are zero.
- But how to choose λ, we will explore this further below.

Another way to write down the Ridge regression problem..

This problem

$$argmin_\beta \underbrace{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2}_{\text{Residual Sum of Squares}} + \underbrace{\lambda \sum_{j=1}^{p} \beta_j^2}_{PenaltyTerm} \quad (3)$$

is the same as

$$min_\beta \underbrace{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k=1}^{p} x_{ij}\beta_j)^2}_{\text{Residual Sum of Squares}} \text{ subject to } \sum_{j=1}^{p} \beta_j^2 \leq s \quad (4)$$

- This is a constrained optimization problem, which you should know from Microeconomics (maximize utility subject to a budget constraint)

Ridge regression is not scale invariant

- *Remember*: The standard OLS coefficient estimates are scale equivariant: multiplying x_j by a constant c simply leads to a scaling of the least squares coefficient estimates by a factor of $1/c$. In other words, regardless of how the jth predictor is scaled, $X_j\hat{\beta}_j$ will remain the same.
- For ridge regression, this is not the case since the β's are directly in the penality term.
- You dont want to penalize coefficients that are large due to the scaling of the underlying x variable.
- We need to make the different β_j's comparable, which we can do easily by standardizing them.

Ridge regression is not scale invariant

- We can scale the x_j by its standard deviation, then all the x_ij values are expressed in terms of standard deviations.
- i.e. the common unit is a standard deviation.
- Divide the value by an estimate of the standard deviation, which is:

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}$$

Some Useful Notation.

An alternative way to express the penalty term is relative to the benchmark OLS, i.e. to normalize the horizontal axis to range from 0 - 1, where 1 corresponds to OLS.

$$\frac{\|\hat{\beta}_\lambda\|_2}{\|\hat{\beta}_{OLS}\|_2} \tag{5}$$

where

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

This is called the $\mathcal{L}_2$ norm, which measures the Euclidian distance from origin. For $p = 2$, this is: $\|\beta\|_2 = \sqrt{\beta_1^2 + \beta_2^2}$.

So when $\lambda \to 0$, $\frac{\|\hat{\beta}_\lambda\|_2}{\|\hat{\beta}_{OLS}\|_2} \to 1$.

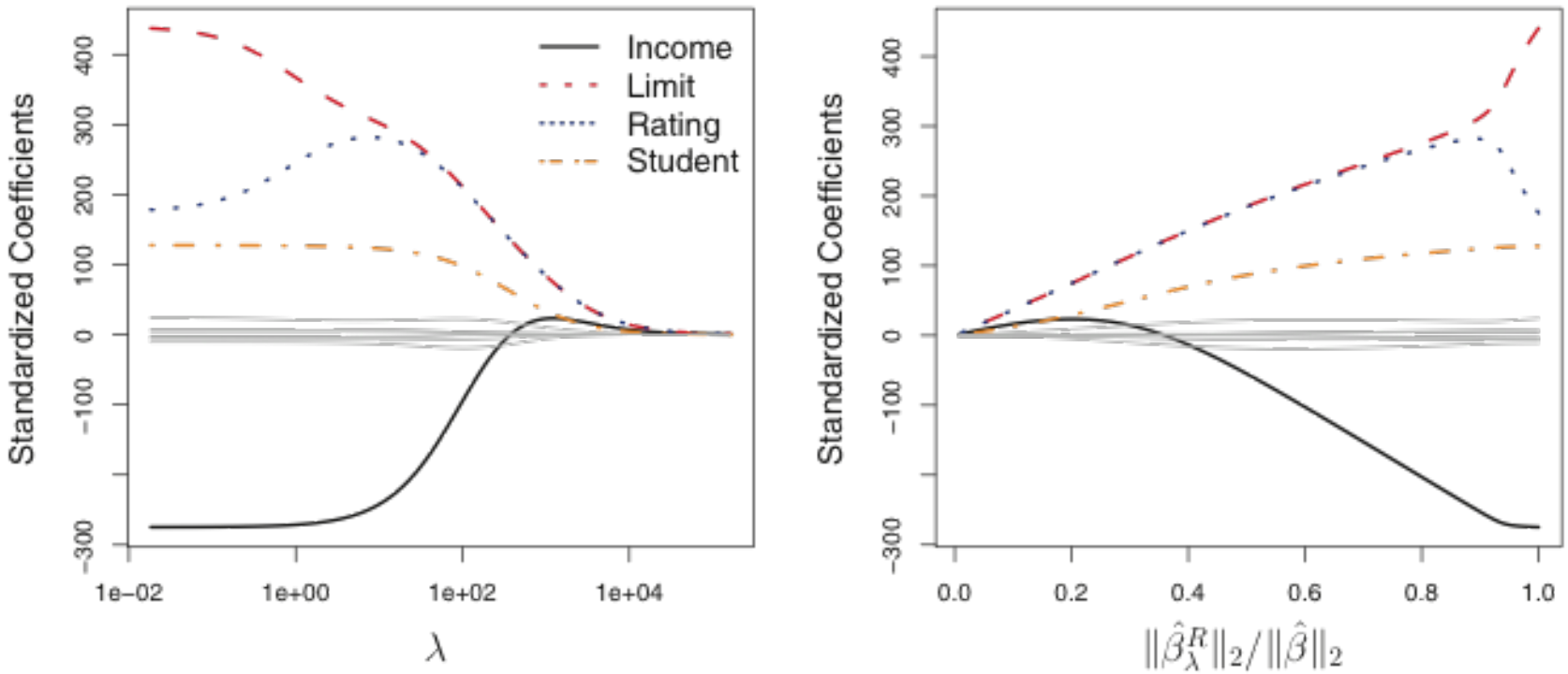

FIGURE 6.4. *The standardized ridge regression coefficients are displayed for the* `Credit` *data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$.

Ok, how we interpret these figures?

- Left figure: Each curve represents the coefficient estimate for a feature as we increase the λ penalty. Note that the grey lines are for variables that are pretty minor. The coeff values shrink towards zero but don't fully make it there.
- Right figure: We are plotting with respect to the $\mathcal{L}_2$ norm. Here, λ starts high and then gets smaller as we move right along the axis.
- Hence the coeffs are very small at the start but then inflate to their unconstrained values as the $\mathcal{L}_2$ norm approaches 1 along the horizontal axis.

What do we gain using Ridge regression?

- The key focus here is the bias variance trade-off.
- Ridge regression, by shrinking coefficients, helps reduce the variance of the prediction and thus, reduces prediction error.
- Roughly, think of these shrunken variables as contributing nuisance variation that doesn't carry well across samples. By down-weighting them we reduce the overall variance of the model.

What do we dislike about Ridge regression?

- As opposed to subset selection, ridge regression does not actually result in simpler models.
- Some estimated coefficients are simply reduced in their absolute value, but we don't get completely rid of irrelevant predictors as we did with subset selection.
- Ideally, we want to allow for coefficients to be exactly equal to zero. This can be achieved, if we make the optimization problem slightly more complicated....

Enter the Lasso

- It stands for: Least Absolute Shrinkage and Selection Operator.
- Lasso shrinks some variables to be exactly equal to zero, rather than just close to zero.
- In particular with a large number of features p, the Lasso can help to significantly reduce the dimensionality of the matrix X.

Setting up the Lasso

The optimization problem is framed as follows:

$$argmin_{\beta} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \mid \beta_j \mid$$

- Again, this can be rewritten as:

$$min_{\beta} \underbrace{\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2}_{\text{Residual Sum of Squares}} \text{ subject to } \sum_{j=1}^{p} \mid \beta_j \mid \leq s \quad (6)$$

- The only thing that changes is the shape of the "circle".
- The constraint in absolute values is also called the $\mathcal{L}_1$ norm of the vector β, i.e.

$$\|\beta\|_1 = \sum_{k=1}^{p} \mid \beta_k \mid$$

A Visual Guide to What Ridge and Lasso Are Doing.

What you need to know in the next picture is:

- The $\hat{\beta}$ is an estimate from unconstrained OLS.
- The red contours represent combinations of β_k values that generate equal levels of RSS. We're only considering $p = 2$ dimensions so our brain can visualize things.
- The green shaded spaces represent the bits of parameter space that are covered by the constraint. The absolute value nature of the Lasso gives us a 'diamond' and the Ridge's square function gives us the 'circle'.

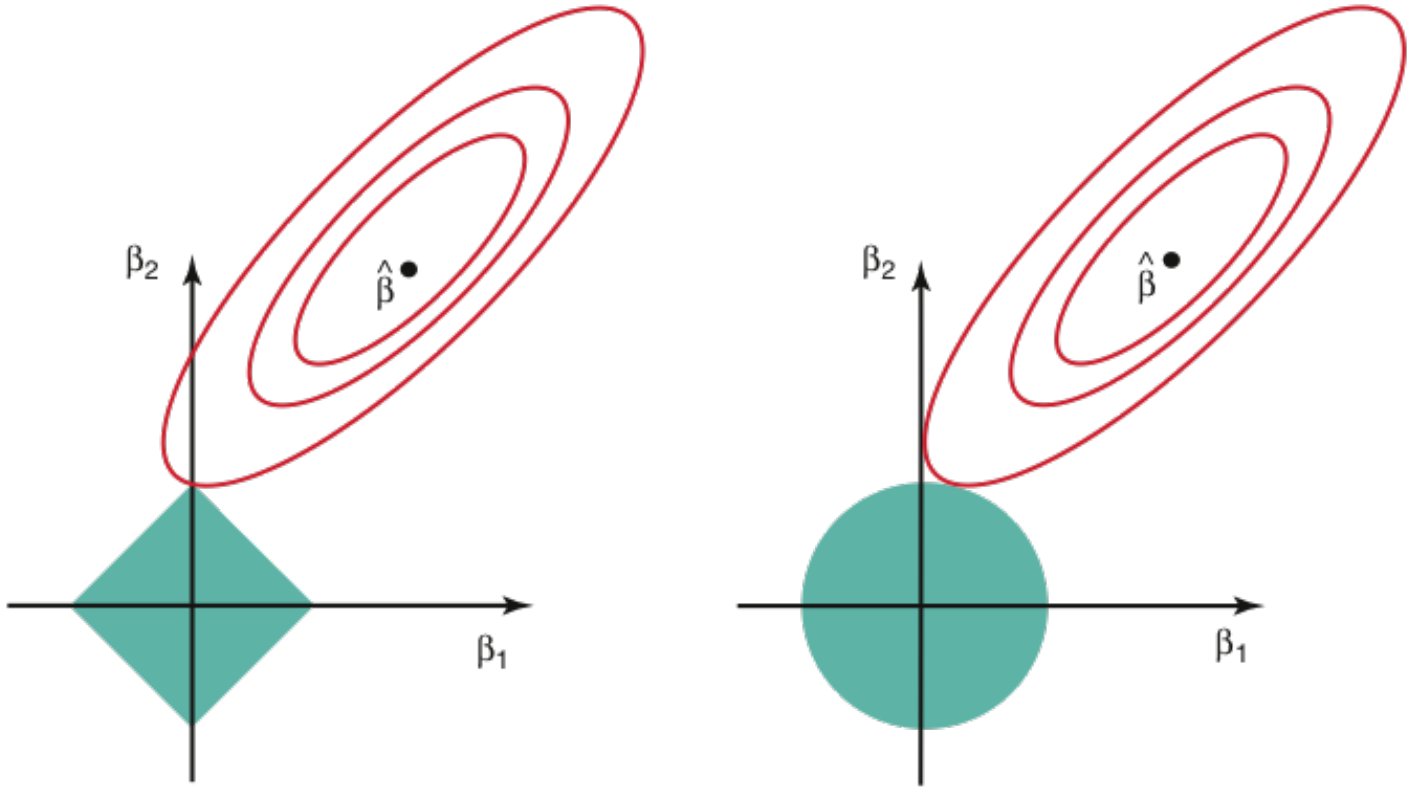

FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

The Point of the Figure

- The key point is that the RSS contours tend to touch the Ridge green region at points where both $\beta_k > 0$ while they touch the Lasso green region at a kink point where (in this case) $\beta_2 = 0$.

- What you have to remember here is the implication that in a high-dimensional p case the Lasso constraint region will have lots more kinks so the chance that we'll get some $\beta_k = 0$ is higher.

- Hence this is why the Lasso is able to exclude some of our p features from its final model.

Lasso or Ridge Regression?

- Neither Ridge regression nor the Lasso will universally dominate the other.
- Lasso performs generally better in a situation, where a relatively small number of predictors have substantial coefficients, and the remaining predictors have coefficients that are very small or that equal zero.
- Ridge regression will perform better when the response is a function of many predictors, the coefficients of which are not very dissimilar from one another.

Lasso or Ridge Regression?

- Lasso and Ridge regression can yield a reduction in variance at the expense of a small increase in bias.
- There are very efficient algorithms for fitting both Ridge and Lasso models; in both cases the entire coefficient paths can be computed with about the same amount of work as a single least squares fit.