# BUSI97287: Advanced Machine Learning

## Regression: Bias-Variance Decomposition and Shrinkage Methods

**Martin Haugh**

Imperial College Business School
Email: martin.b.haugh@gmail.com

Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani (JWHT).

Additional References: Chapter 3 of JWHT and *Learning from Data* by Abu-Mostafa, Lin and Magdon-Ismail.

Required Reading: Sections 6.2 and 6.4 of *ISLR* by JWHT

# Outline

Linear Regression Review
- Potential Problems with Linear Regression

Bias-Variance Decomposition
- A Case Study: Overfitting with Polynomials

The Credit Data-Set from ISLR

Shrinkage Methods
- Ridge Regression
- The Lasso
- Ridge Regression Versus Lasso
- Other Shrinkage Methods

Issues in High Dimensions

# Linear Regression Review

- Linear regression assumes the regression function $\mathbb{E}[Y|\mathbf{X}]$ is linear in the inputs, $X_1, \ldots, X_p$.
- Developed many years ago but still very useful today
  - simple and easy to understand
  - can sometimes outperform more sophisticated models when there is little data available.
- Linear models can also be applied to transformations of the inputs
  - leads to the basis function approach (and kernel regression)
  - which extends the scope of linear models to non-linear models.
- But linear models also have many weaknesses including a tendency to over-fit the data
  - will return to this later when we discuss the bias-variance decomposition and shrinkage methods.

# Linear Regression Review

In linear regression the dependent variable $Y$ is a random variable that satisfies

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \epsilon$$

where $\mathbf{X} = (X_1, \ldots, X_p)$ and $\epsilon$ is the "error" term with $\mathbb{E}[\epsilon] = 0$.

The linear model therefore implicitly assumes that $\mathbb{E}[Y \mid \mathbf{X}]$ is approximately linear in $\mathbf{X} = (X_1, \ldots, X_p)$.

The input or independent variables, $X_i$, are numerical inputs

- or possibly transformations, e.g. product, log, square root, $\phi(x)$, of "original" numeric inputs
- the ability to transform provides considerable flexibility.

The $X_i$'s can also be used as "dummy" variables that encode the levels of qualitative inputs

- an input with $K$ levels would require $K - 1$ dummy variables, $X_1, \ldots, X_{K-1}$

# Model Fitting: Minimizing the Residual Sum of Squares

We are given training data: $(y_1, \mathbf{x}_1), (y_2, \mathbf{x}_2), \ldots, (y_N, \mathbf{x}_N)$.

Then obtain $\hat{\beta}$ by minimizing the residual sum of squares or RSS:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 = \min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

where

$$\mathbf{y} := \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \qquad \mathbf{X} := \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1p} \\ 1 & x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \ldots & x_{Np} \end{bmatrix}$$

# Model Fitting: Minimizing the Residual Sum of Squares

This is a simple (convex) quadratic optimization problem with solution

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

Also have

$$\hat{\mathbf{y}} = \underbrace{\mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top}_{:= \mathbf{H}, \text{ the "hat" matrix}} \mathbf{y} \qquad \text{and} \qquad \hat{\epsilon} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I} - \mathbf{H})\mathbf{y}.$$

Have (implicitly) assumed that $(\mathbf{X}^\top \mathbf{X})$ is invertible.

This is not always the case in which case $\hat{\beta}$ will not be unique

- can resolve by dropping redundant columns from $\mathbf{X}$.

But in many modern applications $p >> N$ in which case at least $N - p$ columns would need to be dropped – something we may not want to do!

- hence the need for another solution approach **e.g.** ridge regression
- for now will assume $p \leq N$.

# Linear Regression with Basis Functions

Can also do everything with basis functions

$$Y = \beta_0 + \sum_{i=1}^{M} \beta_i \psi_i(\mathbf{x}) + \epsilon$$

where $\psi_i : \mathbb{R}^p \mapsto \mathbb{R}$ is the $i^{th}$ basis function.

**e.g.** $\psi_i(\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{p/2}} e^{-\frac{1}{2\sigma^2}\|\mathbf{x}-\boldsymbol{\mu}_i\|_2^2}$.

- $\psi_i(\mathbf{x})$'s are often used to encode domain-specific knowledge.

Parameter estimate:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{\Psi}^\top \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top \mathbf{y}$$

where

$$\boldsymbol{\Psi} = \begin{bmatrix} 1 & \psi_1(\mathbf{x}_1) & \psi_2(\mathbf{x}_1) & \dots & \psi_M(\mathbf{x}_1) \\ 1 & \psi_1(\mathbf{x}_2) & \psi_2(\mathbf{x}_2) & \dots & \psi_M(\mathbf{x}_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \psi_1(\mathbf{x}_N) & \psi_2(\mathbf{x}_N) & \dots & \psi_M(\mathbf{x}_N) \end{bmatrix}$$

# Potential Problems with Linear Regression I

Many problems can arise when fitting a linear model to a particular data-set:

1. Non-linearity of the response-predictor relationships
   - plotting residuals against fitted values are a useful graphical tool for identifying this problem
   - a simple solution is to use non-linear transformations of the predictors.
2. Correlation of error terms
   - a serious problem since estimation of $\sigma^2$ and statistical tests all depend on assumption of zero-correlation
   - problem can arise with time-series data – can detect it then by plotting residuals against time.
3. Non-constant variance or heteroscedasticity of error terms
   - another important assumption that can be tested by plotting residuals against fitted values
   - if problem exists consider applying a concave function to $Y$.
4. Outliers, i.e. points for which $y_i$ is far from the predicted value $\hat{\boldsymbol{\beta}}^\top X_i$
   - could be genuine or a data error
   - may or may not impact fitted model – but regardless will impact $\hat{\sigma}^2$, confidence intervals and p-values, possibly dramatically
   - can identify them by plotting studentized residuals against fitted values – values $> 3$ in absolute value are suspicious.

# Potential Problems with Linear Regression II

5. High-leverage points
   - these are points whose presence has a large impact on the fitted model
   - generally correspond to extreme predictor **X**
   - can identify such points via their leverage statistic, $h_i := H_{ii}$; always the case that $h_i \in [1/N,\ 1]$.

6. Collinearity and multi-collinearity
   - collinearity is the problem when two or more predictor variables are highly correlated
   - difficult then to separate out the individual effects and corresponding coefficients tend to have very high variances
   - can assess multi-collinearity by computing the variance inflation factor (VIF) which is the ratio of Var $\left(\hat{\beta}_i\right)$ when fitting the full model divided by Var $\left(\hat{\beta}_i\right)$ if fit on its own
     - smallest possible value is $1$; rule of thumb is that values exceeding 5 or 10 indicate collinearity
   - solution is to either drop one of the variables or combine them into a single predictor. e.g. in credit data set could combine limit and rating into a single variable.

See discussion in Section 3.3.3 of ISLR for further discussion.

Linear Regression Review
- Potential Problems with Linear Regression

Bias-Variance Decomposition
- A Case Study: Overfitting with Polynomials

The Credit Data-Set from ISLR

Shrinkage Methods
- Ridge Regression
- The Lasso
- Ridge Regression Versus Lasso
- Other Shrinkage Methods

Issues in High Dimensions

# Why Minimize the Sum-of-Squares?

Let $\mathbf{X}$ be non-random and suppose we want to estimate $\theta := \mathbf{a}^\top \boldsymbol{\beta}$.

Then least-squares estimate of $\theta$ is

$$\hat{\theta} = \mathbf{a}^\top \hat{\boldsymbol{\beta}} = \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

– a linear function of the response $\mathbf{y}$.

If the linear model is correct then easy to check that $\mathsf{E}[\hat{\theta}] = \theta$ so $\hat{\theta}$ is unbiased.

**Gauss-Markov Theorem**: Suppose $\mathbf{c}^\top \mathbf{y}$ is any unbiased estimate of $\theta$. Then

$$\mathsf{Var}\left(\mathbf{a}^\top \hat{\boldsymbol{\beta}}\right) \leq \mathsf{Var}\left(\mathbf{c}^\top \mathbf{y}\right).$$

The Gauss-Markov Theorem says that the least-squares estimator has the smallest variance among all linear unbiased estimators.

**Question:** Great! But is unbiasedness a good thing?

# Mean-Squared Error

To answer this question let $\tilde{\theta}$ be some estimator for $\theta$.

The mean-squared-error (MSE) then satisfies

$$\begin{aligned}\mathsf{MSE}(\tilde{\theta}) &= \mathsf{E}\left[\left(\tilde{\theta}-\theta\right)^2\right] \\ &= \mathsf{Var}(\tilde{\theta}) + \underbrace{\left(\mathsf{E}\left[\tilde{\theta}\right]-\theta\right)^2}_{\mathsf{bias}^2}.\end{aligned}$$

If the goal is to minimize MSE then unbiasedness not necessarily a good thing

- can often trade a small increase in bias$^2$ for a larger decrease in variance
- can do with with subset selection methods as well as shrinkage methods
    - an added benefit of some of these methods is improved interpretability.

But first let's study the bias-variance decomposition.

# The Bias-Variance Decomposition

Assume the true model is $Y = f(\mathbf{X}) + \epsilon$ where $\mathsf{E}[\epsilon] = 0$ and $\mathsf{Var}(\epsilon) = \sigma_\epsilon^2$. Let $\hat{f}(\mathbf{X})$ be our estimate at a new fixed point, $\mathbf{X} = \mathbf{x}_0$. Then the error at $\mathbf{x}_0$ assuming the training inputs are fixed, i.e. non-random, is:

$$
\begin{aligned}
\mathsf{Err}(\mathbf{x}_0) &= \mathsf{E}\left[\left(Y_0 - \hat{f}(\mathbf{x}_0)\right)^2\right] \\
&= \mathsf{E}\left[\left(f(\mathbf{x}_0) + \epsilon - \hat{f}(\mathbf{x}_0)\right)^2\right] \\
&= \mathsf{E}\left[\epsilon^2\right] \; + \mathsf{E}\left[\left(f(\mathbf{x}_0) - \hat{f}(\mathbf{x}_0)\right)^2\right] && (1) \\
&= \sigma_\epsilon^2 \; + \; \mathsf{E}\left[\left(f(\mathbf{x}_0) - \mathsf{E}[\hat{f}(\mathbf{x}_0)] + \mathsf{E}[\hat{f}(\mathbf{x}_0)] - \hat{f}(\mathbf{x}_0)\right)^2\right] \\
&= \sigma_\epsilon^2 \; + \; \left(f(\mathbf{x}_0) - \mathsf{E}[\hat{f}(\mathbf{x}_0)]\right)^2 \; + \; \mathsf{E}\left[\left(\hat{f}(\mathbf{x}_0) - \mathsf{E}[\hat{f}(\mathbf{x}_0)]\right)^2\right] \\
&= \sigma_\epsilon^2 + \mathsf{Bias}^2\left(\hat{f}(\mathbf{x}_0)\right) + \mathsf{Var}\left(\hat{f}(\mathbf{x}_0)\right) \\
&= \text{Irreducible Error} + \mathsf{Bias}^2(\mathbf{x}_0) + \mathsf{Variance}(\mathbf{x}_0). && (2)
\end{aligned}
$$

# The Bias-Variance Decomposition

The irreducible error is unavoidable and beyond our control.

But we can exercise control over the bias and variance via our choice of $\hat{f}(\mathbf{x}_0)$

- the more complex the model the smaller the bias and the larger the variance.

**e.g. $k$-Nearest Neighbor Regression**

- Suppose the true model is $y = f(\mathbf{x}) + \epsilon$ with $\mathsf{E}[\epsilon] = 0$ and $\mathsf{Var}(\epsilon) = \sigma_\epsilon^2$.
- Have independent training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

In $k$-nearest neighbor regression our prediction at a new point $\mathbf{x}_0$ is

$$\hat{f}(\mathbf{x}_0) = \frac{1}{k} \sum_{l=1}^{k} y_{(l)}$$

where $\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(k)}$ are the $k$ nearest neighbors to $\mathbf{x}_0$.

# The Bias-Variance Decomposition

## e.g. $k$-Nearest Neighbor Regression (ctd.)

In this case (2) reduces to

$$\text{Err}(\mathbf{x}_0) = \underbrace{\sigma_\epsilon^2}_{\text{Irreducible}} + \underbrace{\left( f(\mathbf{x}_0) - \frac{1}{k} \sum_{l=1}^{k} f(\mathbf{x}_{(l)}) \right)^2}_{\text{Bias}^2} + \underbrace{\frac{\sigma_\epsilon^2}{k}}_{\text{Variance}}$$

Easily see that $\text{E}[\hat{f}(\mathbf{x}_0)] = \frac{1}{k} \sum_{l=1}^{k} f(\mathbf{x}_{(l)})$ so bias term is correct.

**Question:** Show that the variance term is correct.

Here $k$ is inversely related to model "complexity" (why?) and we see:

- Bias typically decreases with model complexity
- Variance increases with model complexity

# Example: the Bias-Variance Trade-Off

Consider the following example from Bishop:

1. The "true" model to be estimated is

$$y(x) = \sin(2\pi x) + \epsilon, \quad x \in [0, 1], \qquad \epsilon \sim \mathsf{N}(0, c) \tag{3}$$

   - a very nonlinear function of $x$.

2. We fit a linear regression model with $M = 24$ Gaussian basis functions

$$\psi_j(x) := e^{-\frac{1}{2\sigma^2}(x-\mu_j)^2}$$

   with $\mu_j = \frac{j}{M-1}$ for $j = 0, \ldots, M-1$ and $\sigma = \frac{1}{M-1}$.

3. Including the constant term, the parameter vector $\boldsymbol{\beta}$ is $(M+1) \times 1$.

4. Will also include a regularization term so that regression problem solves

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{j=1}^{N} \left( Y_j - \beta_0 - \sum_{i=1}^{M} \beta_i \psi_i(\mathbf{x}_j) \right)^2 + \frac{\lambda}{2} \boldsymbol{\beta}^\top \boldsymbol{\beta} \tag{4}$$

# Example: the Bias-Variance Trade-Off

5. A data-set is of the form $\mathcal{D} = \{(y_i, x_i) : i = 1, \ldots, N\}$, with $N = 25$
   - the $x_i$'s are sampled randomly from $[0, 1]$
   - the $y_i$'s are then sampled using (3).
     - so noise comes both from measurement and sampling.
6. We generate $L = 100$ of these data-sets.
7. The model is fit by solving (4) for each of the $L$ data-sets and various values of $\lambda$.

Results are displayed in Figure 3.5:

The model bias is clear from graphs in right-hand column.

The variance of individual fits is clear from graphs in left-hand column.

The bias-variance tradeoff is clear and quantified in Figure 3.6.

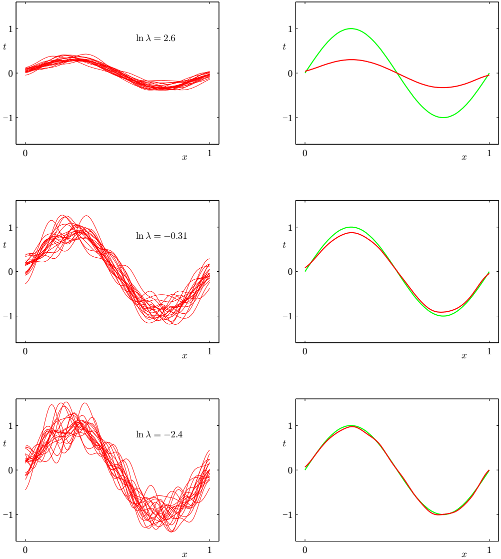


**Figure 3.5 from Bishop**: Illustration of the dependence of bias and variance on model complexity, governed by a regularization parameter $\lambda$, using the sinusoidal data set from Chapter 1. There are $L = 100$ data sets, each having $N = 25$ data points, and there are $24$ Gaussian basis functions in the model so that the total number of parameters is $M = 25$ including the bias parameter. The left column shows the result of fitting the model to the data sets for various values of $\ln \lambda$ (for clarity, only $20$ of the $100$ fits are shown). The right column shows the corresponding average of the $100$ fits (red) along with the sinusoidal function from which the data sets were generated (green).

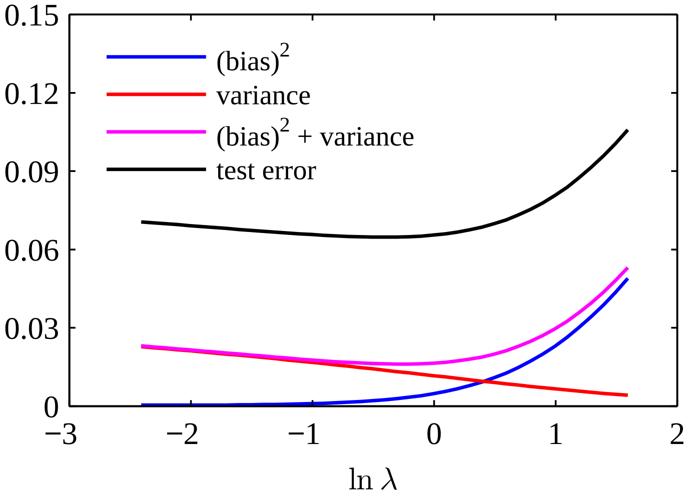


**Figure 3.6 from Bishop**: Plot of squared bias and variance, together with their sum, corresponding to the results shown in Figure 3.5. Also shown is the average test set error for a test data set size of $1000$ points. The minimum value of $(\text{bias})^2 +$ variance occurs around $\ln\lambda = -0.31$, which is close to the value that gives the minimum error on the test data.

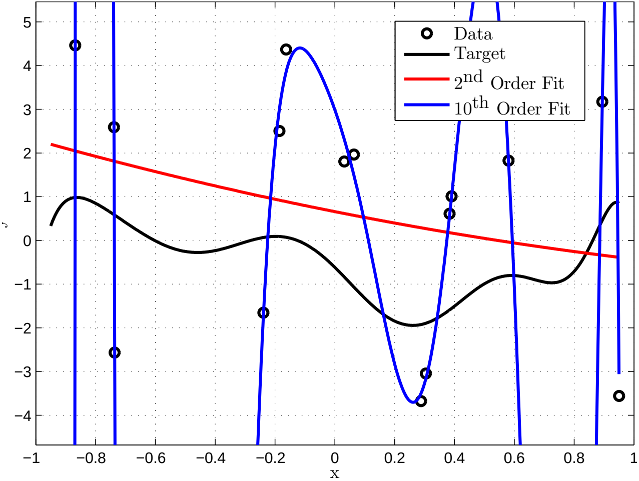
# A Case Study: Overfitting with Polynomials

Overfitting is a very serious issue that needs to be handled in supervised learning problems.

To explore overfitting in further detail we will consider two $1$-dimensional polynomial regression problems.

### Problem 1

- True model is $y = f(x) + \epsilon$ where $\epsilon$ is IID noise and $f(x)$ is a $10^{th}$ order polynomial on $x \in \mathbb{R}$.
- There are $n = 15$ data-points: $(x_1, y_1), \ldots, (x_n, y_n)$
  - the $x_i$'s were generated $\sim U(-1, 1)$ and then $y_i = f(x_i) + \epsilon_i$ where the $\epsilon_i$'s were generated IID $\mathrm{N}(0, 3)$.
- We fit $2^{nd}$ and $10^{th}$ order polynomials to this data via simple linear regression, that is we regress $Y$ on $1, X, \ldots, X^J$ where $J = 2$ or $J = 10$.
- Results displayed in the figure on the next slide.



**Fitting a Low-Order Polynomial With Noisy Data:** The target curve is the $10^{th}$ order polynomial $y = f(x)$.

# A Case Study: Overfitting with Polynomials

**Question:** Which regression results in a superior fit *to the data*?

**Question:** Which regression results in a superior *out-of-sample* or generalization error?

Note that the set of $10^{th}$ order polynomials *contains* the true target function, $y = f(x)$, whereas the set of $2^{nd}$ order polynomials does not.

We might therefore expect the $10^{th}$ order fit to be superior to the $2^{nd}$ order fit

- but this is not the case!

**Question:** Why do you think the $2^{nd}$ order fit does a better job here?

**Question:** Do you think the $2^{nd}$ order fit will always be better irrespective of $N$, the number of data-points?

# A Case Study: Over-fitting with Polynomials

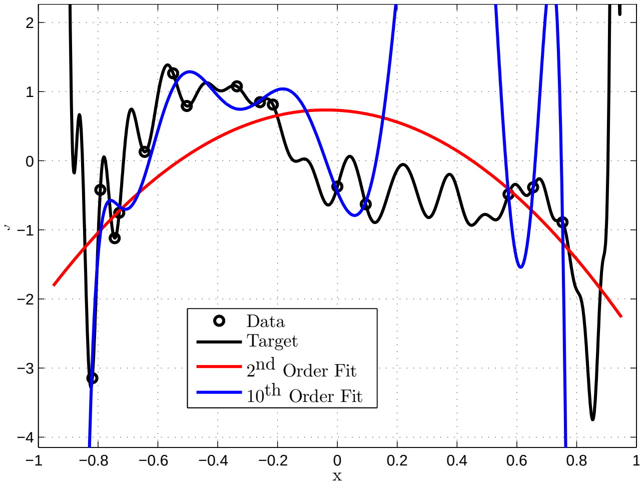
### Problem 2

- True model is $y = f(x)$ and $f(x)$ is a $50^{th}$ order polynomial on $x \in \mathbb{R}$.
- There are $n = 15$ data-points: $(x_1, y_1), \ldots, (x_n, y_n)$
  - the $x_i$'s were generated $\sim U(-1, 1)$ and then $y_i = f(x_i)$ so the observations are noiseless.
- We fit $2^{nd}$ and $10^{th}$ order polynomials to this data via simple linear regression, that is we regress $Y$ on $1, X, \ldots, X^J$ where $J = 2$ or $J = 10$.
- The results are displayed in the figure on the next slide.

Commonly thought that over-fitting occurs when the fitted model is too complex relative to the true model

- but this is not the case here: clearly the $10^{th}$ order regression over-fits the data but a $10^{th}$ order polynomial is considerably less complex than a $50^{th}$ order polynomial.

What matters is how the model complexity matches the quantity and quality of the data, not the (unknown) target function.



**Fitting a High-Order Polynomial With Noiseless Data:** The target curve is the $10^{th}$ order polynomial $y = f(x)$.

Note: This case study is based on the case study in Section 4.1 of "*Learning from Data*" by Abu-Mostafa, Magdon-Ismail and Lin.

# Methods for Exploring the Bias-Variance Trade-Off and Controlling Overfitting

**Vital** then to control over-fitting when performing supervised learning, i.e. regression or classification. There are many approaches:

- Subset selection where we retain only a subset of the independent variables
- Shrinkage methods where coefficients are shrunk towards zero.
- Regularization where we penalize large-magnitude parameters
  - shrinkage often achieved via regularization

Cross-validation often used to select the specific model. Other methods include:

- Bayesian models
  - many shrinkage / regularization methods can be interpreted as Bayesian models where the penalty on large-magnitude parameters becomes a prior distribution on those parameters.
- Methods that explicitly penalize the number of parameters, $p$, in the model
  - Akaike Information Criterion (AIC) $= -2\ln(\text{likelihood}) + 2(p+1)$
  - Bayesian Information Criterion (BIC): $-2\ln(\text{likelihood}) + (p+1)\ln(N)$

  Choose the model that minimizes the AIC or BIC
  - these methods apply to models fit via MLE.

Linear Regression Review
    Potential Problems with Linear Regression

Bias-Variance Decomposition
    A Case Study: Overfitting with Polynomials

The Credit Data-Set from ISLR

Shrinkage Methods
    Ridge Regression
    The Lasso
    Ridge Regression Versus Lasso
    Other Shrinkage Methods

Issues in High Dimensions

# The Credit Data-Set from ISLR

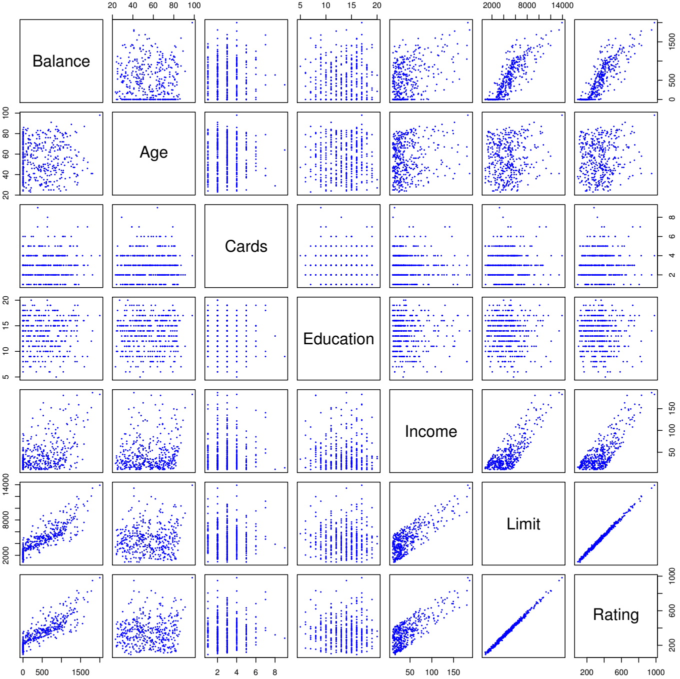
The credit data-set from ISLR contains quantitative data on following variables for a number of customers. See Fig. 3.6 for corresponding scatter-plot matrix.

- balance = average credit card debt - this is the dependent variable
- age (in years)
- cards (number of credit cards)
- education (years of education)
- income (in thousands of dollars)
- limit (credit limit)
- rating (credit rating)

There are also four qualitative variables:

- gender
- student (student status)
- status (marital status)
- ethnicity (Caucasian, African American or Asian)

See Section 3.3 of ISLR for analysis and discussion of this data-set and in particular, how to handle qualitative variables using dummy variables.


Balance
Age
Cards
Education
Income
Limit
Rating
20
40
60
80
100
5
10
15
20
2000
8000
14000
0
500
1500
2
4
6
8
50
100
150
200
600
1000

Linear Regression Review

- Potential Problems with Linear Regression

Bias-Variance Decomposition

- A Case Study: Overfitting with Polynomials

The Credit Data-Set from ISLR

Shrinkage Methods

- Ridge Regression
- The Lasso
- Ridge Regression Versus Lasso
- Other Shrinkage Methods

Issues in High Dimensions

# Shrinkage Methods

Will focus mainly on two shrinkage methods:

1. Ridge regression where we solve:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\beta \right\|^2 + \lambda \cdot \frac{1}{2} \left\| \boldsymbol{\beta} \right\|_2^2 \right\}.$$

2. The *Least Absolute Shrinkage and Selection Operator* or Lasso solves

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^2 + \lambda \left\| \boldsymbol{\beta} \right\|_1 \right\} \quad \left\| \boldsymbol{\beta} \right\|_1 = \sum_{j=1}^{n} |\beta_j|$$

   As $\lambda$ increases, coefficients will abruptly drop to zero.

**Question:** How should we choose $\lambda$?

**Note:** Shrinkage methods can also be applied to classification problems!

# Ridge Regression

Ridge regression solves

$$\hat{\boldsymbol{\beta}}^{\mathsf{R}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^2 + \frac{\lambda}{2} \sum_{j=1}^{p} \beta_j^2 \right\}$$

- shrinks regression coefficients towards $0$ by imposing a penalty on their size
- $\lambda$ is a complexity parameter that controls the amount of shrinkage.

An equivalent formulation is

$$\begin{aligned} \hat{\boldsymbol{\beta}}^{\mathsf{R}} &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^2 \right\} \qquad (5) \\ \text{subject to} \quad & \sum_{j=1}^{p} \beta_j^2 \leq s \end{aligned}$$

It is standard (why?) to scale and standardize inputs before applying ridge regression.

# Ridge Regression

Note $\beta_0$ is generally not shrunk so the procedure does not depend on origin chosen for $Y$.

To handle this and use matrix notation we can split estimation into two steps:

1. Set $\hat{\beta}_0 = \bar{y} = \frac{\sum_{i=1}^{N} y_i}{N}$
2. Center the inputs so that $x_{ij} \rightarrow x_{ij} - \bar{x}_j$.
   Now estimate $\beta_1, \ldots, \beta_p$ using ridge regression without intercept and using the centered $x_{ij}$'s.

Dropping $\beta_0$ from $\boldsymbol{\beta}$, the ridge regression of step 2 therefore solves

$$\hat{\boldsymbol{\beta}}^{\mathsf{R}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \frac{\lambda}{2} \boldsymbol{\beta}^\top \boldsymbol{\beta} \right\}$$

which has solution

$$\hat{\boldsymbol{\beta}}^{\mathsf{R}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}. \tag{6}$$

# Ridge Regression

Note that $\hat{\beta}^{\mathsf{R}}$ is obtained as the solution of a least squares problem except that a positive term, i.e. $\lambda$, has been added to the diagonal of $\mathbf{X}^\top\mathbf{X}$

- this makes the problem non-singular, even if $\mathbf{X}^\top\mathbf{X}$ does not have full rank
- this was the main motivation for ridge regression when first introduced.

Figure 6.4 from ISLR displays $\hat{\beta}^{\mathsf{R}}$ for various values of $\lambda$ and $||\hat{\beta}^{\mathsf{R}}_\lambda||_2/||\hat{\beta}||_2$

- can interpret $||\hat{\beta}^{\mathsf{R}}_\lambda||_2/||\hat{\beta}||_2$ as a measure of the total shrinkage achieved
- note that we recover the least squares solution as $\lambda \to 0$.

# Ridge Regression on the Credit Data Set

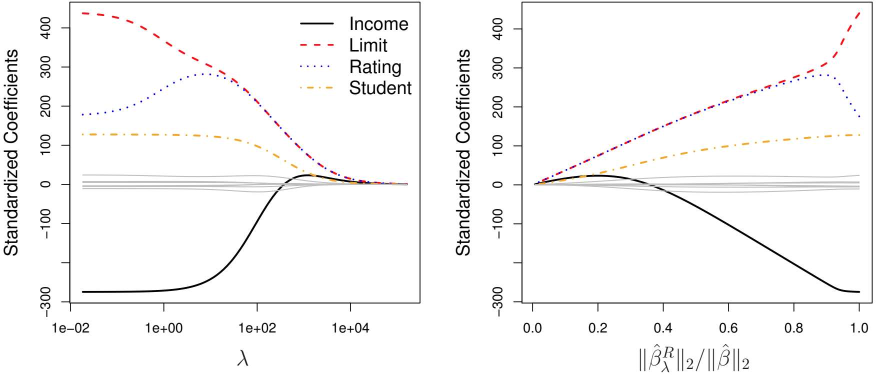


**Figure 6.4 from ISLR**: The standardized ridge regression coefficients are displayed for the Credit data set, as a function of $\lambda$ and $||\hat{\boldsymbol{\beta}}^{\mathrm{R}}_\lambda||_2/||\hat{\boldsymbol{\beta}}_\lambda||_2$.

Note that as $\lambda$ increases coefficients are shrunk towards zero.

Also note that coefficients are generally non-zero for any value of $\lambda$
- so ridge regression does not result in sparse models.

# Selecting $\lambda$ Via Cross-Validation

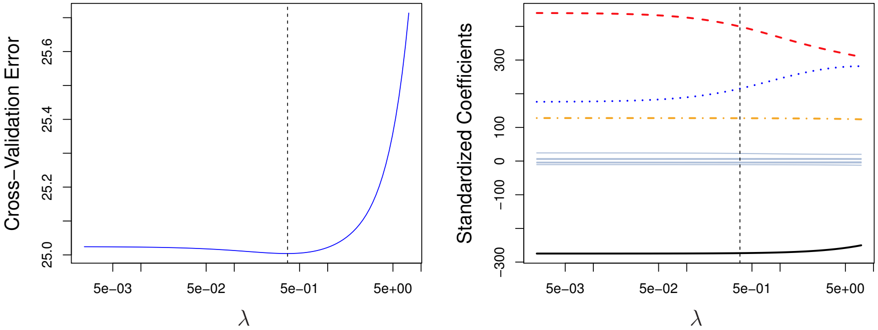


**Figure 6.12 from ISLR**: Cross-validation errors that result from applying ridge regression to the Credit data set with various value of $\lambda$. Right: The coefficient estimates as a function of $\lambda$. The vertical dashed lines indicate the value of $\lambda$ selected by cross-validation.

Using cross-validation to select $\lambda$ for the Credit data set results in only a modest amount of shrinkage.

And the cv error is relatively insensitive to choice of $\lambda$ here
- so little improvement over least squares solution.

# Why Does Ridge Regression Improve Over Least Squares?

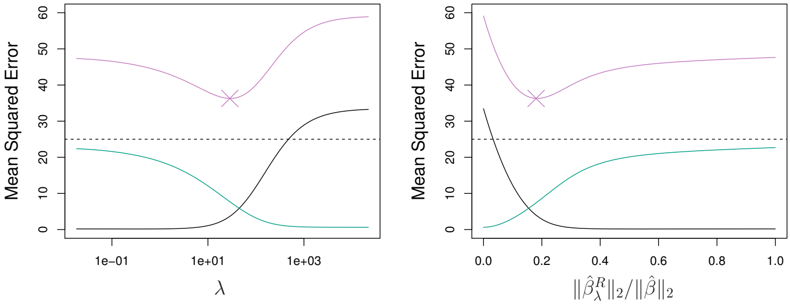


**Figure 6.5 from ISLR**: Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of $\lambda$ and $||\hat{\boldsymbol{\beta}}^R_\lambda||_2/||\hat{\boldsymbol{\beta}}_\lambda||_2$. The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.

Ridge regression (and Lasso) often (significantly) outperform least-squares because it is capable (through selection of $\lambda$) of trading off a small increase in bias for a potentially much larger decrease in variance.

# The Lasso

Recall that the Lasso solves

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^2 + \lambda \left\| \boldsymbol{\beta} \right\|_1 \right\}$$

where $\|\boldsymbol{\beta}\|_1 := \sum_{j=1}^{n} |\beta_j|$.

Penalizing the 1-norm ensures that coefficients will abruptly drop to zero as $\lambda$ increases – results in superior interpretability.

The Lasso can also be formulated by constraining $\|\boldsymbol{\beta}\|_1$:

$$\hat{\boldsymbol{\beta}}^{\mathsf{L}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^2 \right\} \quad (7)$$

$$\text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s$$

Unlike ridge regression, a closed-form solution is not available for the Lasso

- but it can be formulated as a convex quadratic optimization problem and is therefore easy to solve numerically.

# Lasso on the Credit Data Set

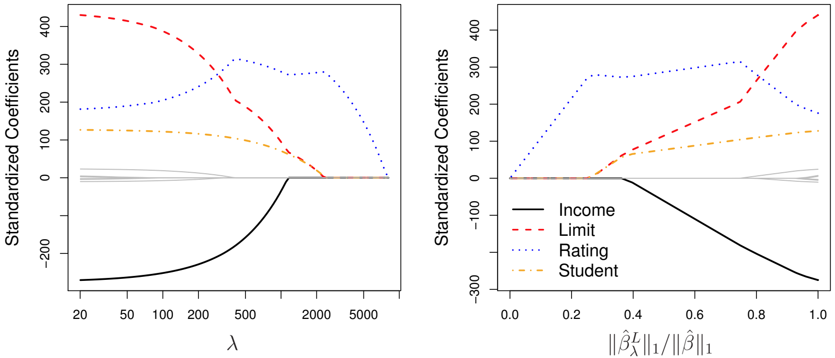


**Figure 6.6 from ISLR**: The standardized lasso coefficients on the Credit data set are shown as a function of $\lambda$ and $||\hat{\boldsymbol{\beta}}^L_\lambda||_1/||\hat{\boldsymbol{\beta}}_\lambda||_1$.

Note how coefficients abruptly drop to 0 as $\lambda$ increases in Figure 6.6

- contrast this with ridge regression!

Lasso results in sparse models then and can be viewed as a method for subset selection.

# A Simulated Data Set

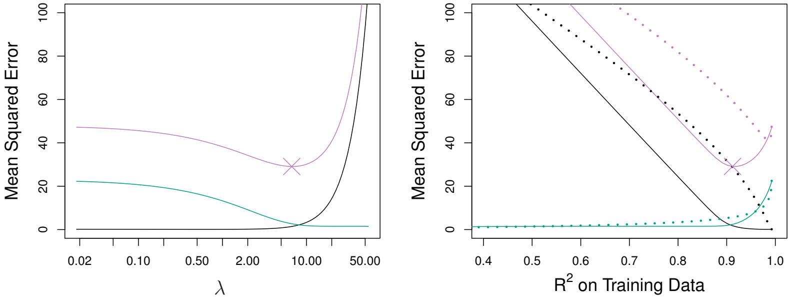


**Figure 6.9 from ISLR**: Left: Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso. The simulated data is similar to that in Figure 6.8, except that now only two predictors are related to the response. Right: Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dashed). Both are plotted against their $R^2$ on the training data, as a common form of indexing. The crosses in both plots indicate the lasso model for which the MSE is smallest.

Figure 6.9 displays results from a simulated data set with $p = 45$ predictors – but the response $Y$ is a function of only $2$ of them!

# Selecting $\lambda$ Via Cross-Validation

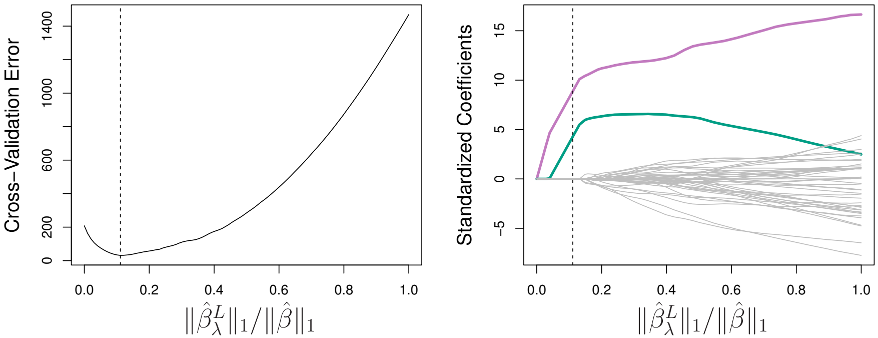


**Figure 6.13 from ISLR**: Left: Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data set from Figure 6.9. Right: The corresponding lasso coefficient estimates are displayed. The vertical dashed lines indicate the lasso fit for which the cross-validation error is smallest.

Note how the optimal $\lambda$ (chosen via cross-validation) correctly identifies the model with the 2 predictors

- contrast with least squares solution at far right of right-hand figure!

# Lasso Versus Ridge Regression

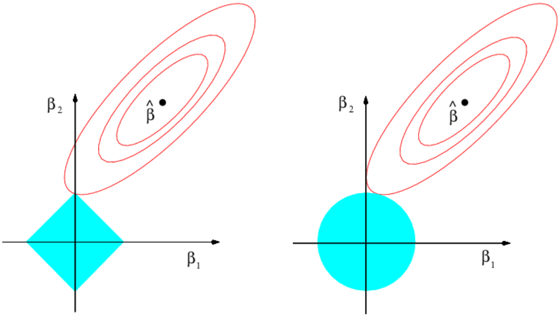


**Figure 6.7 from ISLR**: Contours of the error and constraint functions for the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$, while the red ellipses are the contours of the RSS.

Contours of the error and constraint functions of the formulations in (5) and (7) are displayed in Figure 6.7.

This perspective makes it clear why Lasso results in a sparse solution whereas ridge regression does not.

# Ridge Regression Versus Lasso

The following e.g. (taken from ISLR) provides further intuition for why Lasso results in sparse solutions and ridge regression does not. We assume:

- $N = p$.
- **X** is a diagonal matrix with $1$'s on the diagonal.
- There is no intercept term.

Least squares then solves $\min_{\beta_1,\ldots,\beta_p} \sum_{j=1}^{N} (y_j - \beta_j)^2$

Solution is $\hat{\beta}_j = y_j$.

Ridge regression solves $\min_{\beta_1,\ldots,\beta_p} \sum_{j=1}^{N} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$

Can check solution is $\hat{\beta}_j^R = y_j/(1+\lambda)$.

Lasso solves $\min_{\beta_1,\ldots,\beta_p} \sum_{j=1}^{N} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$

Can check solution is

$$\hat{\beta}_j^L = \begin{cases} y_j - \lambda/2, & \text{if } y_j > \lambda/2; \\ y_j + \lambda/2, & \text{if } y_j < -\lambda/2; \\ 0, & \text{if } |y_j| \le \lambda/2. \end{cases}$$

# Other Shrinkage Methods

Group Lasso:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\beta \right\|^2 + \lambda \sum_{k=1}^{m} \left\| \boldsymbol{\beta}_k \right\|_2 \right\}$$

where $\boldsymbol{\beta}_k$ are non-overlapping sub-vectors of $(\beta_1, \ldots, \beta_p)^\top$

- Induces all the coefficients in the sub-vector to go to zero
- Useful when there are dummy variables in the regression.

Composite norm methods:

$$\min_{\boldsymbol{\beta}} \left\{ \left\| \mathbf{y} - \mathbf{X}\beta \right\| + \lambda \sum_{k=1}^{m} \left\| \boldsymbol{\beta}_k \right\|_2 \right\}$$

- Useful when we want to force $\mathbf{X}\beta = \mathbf{y}$.

Elastic nets:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\beta \right\|^2 + \lambda \big( (1-\alpha) \frac{1}{2} \left\| \boldsymbol{\beta} \right\|_2^2 + \alpha \left\| \boldsymbol{\beta} \right\|_1 \big) \right\}$$

Linear Regression Review

- Potential Problems with Linear Regression

Bias-Variance Decomposition

- A Case Study: Overfitting with Polynomials

The Credit Data-Set from ISLR

Shrinkage Methods

- Ridge Regression
- The Lasso
- Ridge Regression Versus Lasso
- Other Shrinkage Methods

Issues in High Dimensions

# High Dimensional Problems

Traditionally problems in statistics were low-dimensional with $p < N$ and often $p << N$.

But many modern settings have $p > N$. For example:

1. Classical statistics might attempt to predict blood pressure as a function of age gender and body-mass-index (BMI). Modern methods might also use measurements for approx 500k single nucleotide polymorphisms (SNPs).
2. Online advertisers may want to predict the purchasing behavior of someone using a search engine. Dummy variables for each of $p$ search terms might be included as predictors with $p_i = 1$ if the $i^{th}$ term was previously searched by the user and $p_i = 0$ otherwise.
3. Speech recognition problems where we have speech samples for $N$ speakers. To represent a speech sample as a numeric vector we require very large $p$.

Need to be very careful in these high-dimensional settings where (unique) least squares solutions do not even exist.

Even if $p$ is smaller than but still close to $N$ then similar problems still arise.

Similar observations hold true for classification problems that use classical approaches such as LDA, QDA, logistic regression etc.

# Issues in High Dimensions

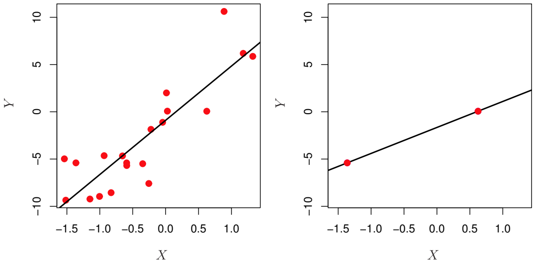


**Figure 6.22 from ISLR**: Left: Least squares regression in the low-dimensional setting. Right: Least squares regression with $n = 2$ observations and two parameters to be estimated (an intercept and a coefficient).

Problem in Fig. 6.22 is low dimensional but demonstrates what can go wrong when we have too little data relative to problem dimension

- this certainly occurs when $p \approx N$
- saw similar issues with earlier case-study.

When $p \geq N$ least squares can fit the data perfectly and so $R^2$ will equal 1

- but likely that massive over-fitting is taking place.

# Issues in High Dimensions

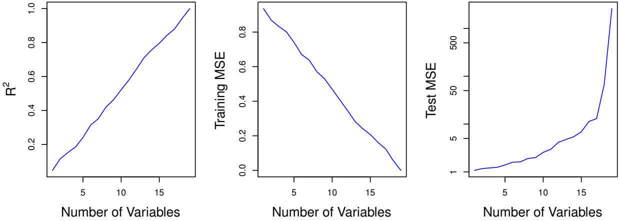


**Figure 6.23 from ISLR**: On a simulated example with $n = 20$ training observations, features that are completely unrelated to the outcome are added to the model. Left: The $R^2$ increases to 1 as more features are included. Center: The training set MSE decreases to 0 as more features are included. Right: The test set MSE increases as more features are included.

Note that in Figure 6.23 the features are completely unrelated to the response!

Estimating test error is therefore particularly vital in these settings – but $C_p$, AIC and BIC are not suitable due to difficulty in estimating $\sigma^2$.

The solution is to restrict the choice of models which is exactly what subset selection, ridge regression, lasso etc. do.

# Issues in High Dimensions

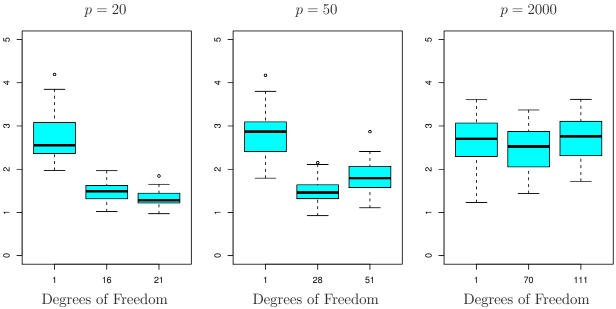


**Figure 6.24 from ISLR**: The lasso was performed with $n = 100$ observations and three values of $p$, the number of features. Of the $p$ features, 20 were associated with the response. The boxplots show the test MSEs that result using three different values of the tuning parameter $\lambda$ in (6.7). For ease of interpretation, rather than reporting $\lambda$, the degrees of freedom are reported; for the lasso this turns out to be simply the number of estimated non-zero coefficients. When $p = 20$, the lowest test MSE was obtained with the smallest amount of regularization. When $p = 50$, the lowest test MSE was achieved when there is a substantial amount of regularization. When $p = 2,000$ the lasso performed poorly regardless of the amount of regularization, due to the fact that only 20 of the 2,000 features truly are associated with the outcome.

# Issues in High Dimensions

Note results in Figure 6.24 where only 20 features were relevant.

Degrees-of-freedom, df($\lambda$), is reported instead of $\lambda$

- df($\lambda$) $=$ number of non-zero coefficient estimates in the lasso solution
- much easier to interpret!

When $p = 20$ or $p = 50$ we see the importance of choosing a good value of $\lambda$.

But we also see that lasso performed poorly when $p = 2000$

- because test error tends to increase with $p$ unless the new features are actually informative
- note the implications of this observation – there is a cost to be paid for blindly adding new features to a model even when regularization is employed!

Multi-collinearity is clearly present in high-dimensional problems – therefore cannot hope to identify the very best predictors

- instead hope to identify good predictors.

Note that linear models – which we have been considering – are generally popular for high dimensional problems. Why?