Lecture 7

Regularization

ISLR 6, ESL 3

Jilles Vreeken
Aleksandar Bojchevski

UNIVERSITÄT DES SAARLANDES

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Introduction

We can **expand on the basic linear model** in several ways

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

We can make it **more flexible**

- add in **nonlinear basis functions** (Chapter 7)
- **leave the linear paradigm** altogether (Chapter 8)

We can make it **less flexible**

- **subset selection**: only use a subset of the variables in the model
- **shrinkage**: penalize models with large or with many non-zero coefficients
- **dimensionality reduction**: projecting the data into a low-dimensional subspace

Why Simpler Models?

Gauss-Markov Theorem

- the full linear model is the unbiased linear model with the smallest variance

Recall that we have the bias-variance tradeoff as

$$E\left[y_0 - \hat{f}(x_0)\right]^2 = Var\left(\hat{f}(x_0)\right) + \left[Bias\left(\hat{f}(x_0)\right)\right]^2 + Var(\epsilon)$$

- we can often strongly reduce variance at a negligible increase in bias by constraining the coefficients

Furthermore

- if $p > n$ there is no longer a unique least-squares estimate
- selecting a small subset of the coefficients makes the model more interpretable

Best Subset Selection

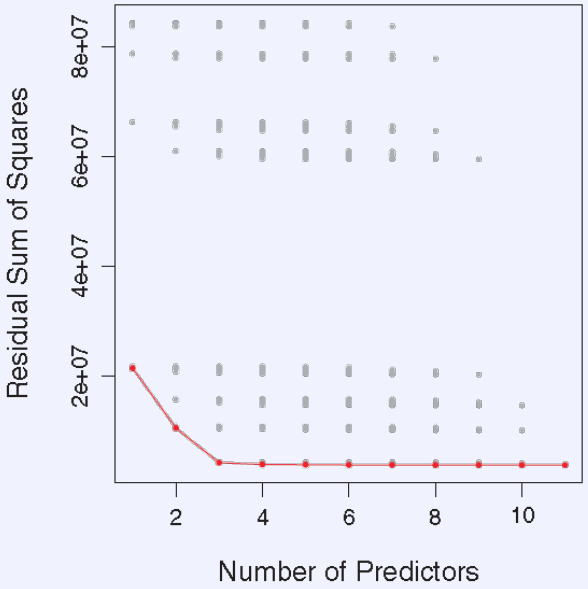
Find the best model for every possible subset of predictors

- there are 2^p such models
- assess the test error of each of these models, and then choose the best
- this scales to $p \approx 30$

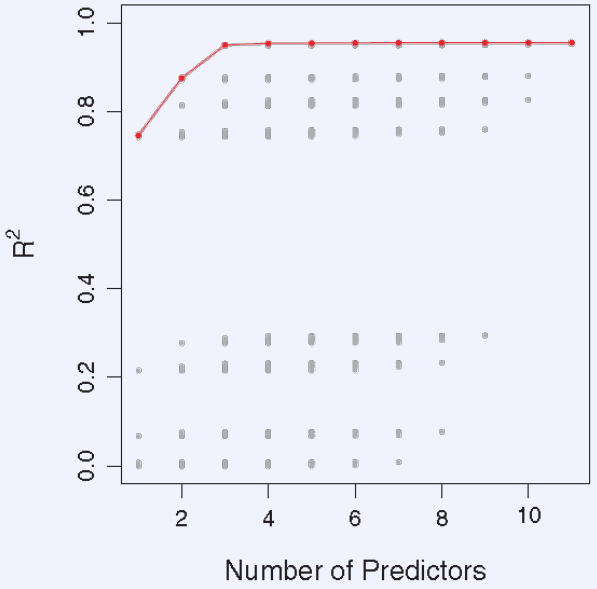
There exist various methods for assessing test error

- we know cross-validation, which is based on resampling
- there also exist formulas that estimate test error in terms of training error plus a corrective term: $AIC, BIC, adjusted\ R^2$

Example Best Subset Selection



Best subset selection on the Credit data
Training error measured via RSS



Best subset selection on the Credit data
Training error measured via R^2

Stepwise Selection

Greedy approach

1. start with a null model $\mathcal{M}_0$ that consists of only the intercept
2. iteratively add that predictor that improves the model most* yielding model $\mathcal{M}_i$
3. choose the best model among $\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_p$ using some method for assessing test error

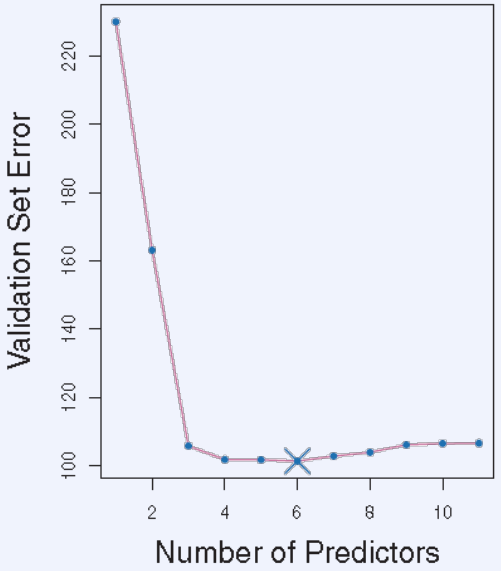
- only $p(p+1)/2$ models need to be calculated
- in step 2 we can assess training error as we compare models of the same number of variables
- in step 3 we have to assess test error as we compare models with different numbers of variables.

- backward stepwise selection works similarly: start with full linear model, incrementally eliminate variables
- in each step the variable is chosen whose elimination deteriorates the model least
- hybrid approaches allow for switching between forward and backward steps

* in terms of smallest RSS or largest R^2 (ISSL 5.2)

Choosing the Optimal Model

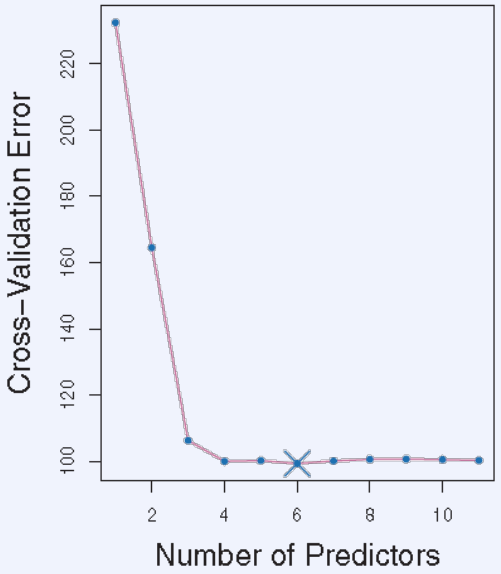
1. Validation set
   - **Example** three fourths for training, one fourth for testing



Choosing the Optimal Model

2. Cross-validation

- here, 10-folds
- the curve is very flat for more than three predictors
- we likely only pick up noise $p > 3$
- it is not useful to choose the 'best' model



One-standard-error rule:

Choose the simplest model within one standard error of the best model

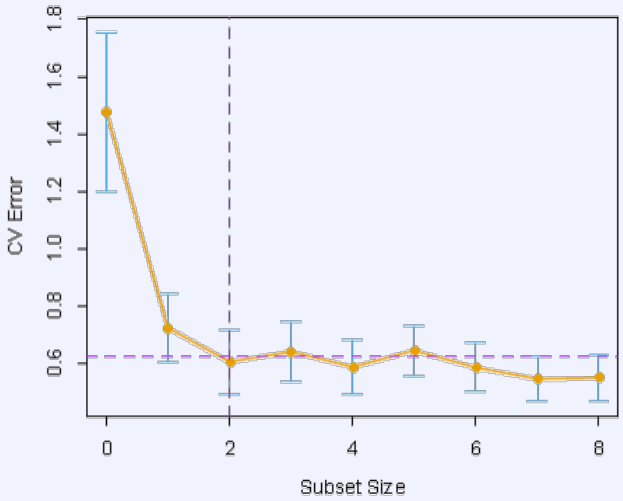
Choosing the Optimal Model

2. Cross-validation

- here, 10-folds
- the curve is very flat for more than three predictors
- we likely only pick up noise $p > 3$
- it is not useful to choose the 'best' model

One-standard-error rule:

Choose the simplest model within one standard error of the best model



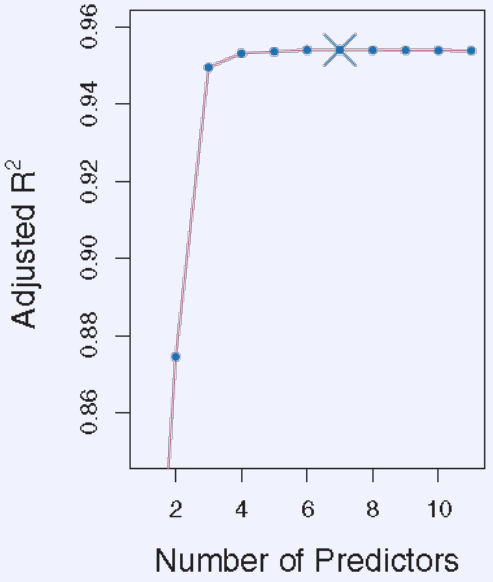
Application of the one-standard-error-rule on another dataset (ESL p 62)

Choosing the Optimal Model

3. Adjusted R^2

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

- vanilla R^2 monotonically increases with the number of variables
- the adjustment counteracts this
- maximizing adjusted R^2 is the same as minimizing $RSS/(n-d-1)$
- rationale: after all informative variables are in the model, including additional noise variables will decrease RSS but not $RSS/(n-d-1)$
- adjusted R^2 does not have a sound statistical foundation



Adjusted R^2 on the Credit data set
The best model involves the variables
income, limit, rating, cards, age, student, gender

Choosing the Optimal Model

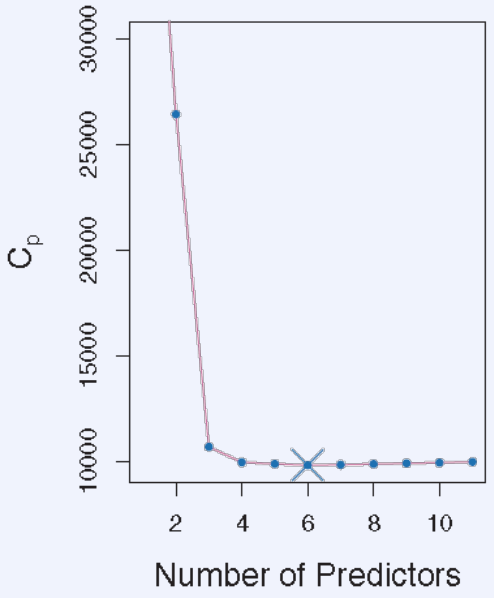
4. the C_p statistic (for least-square models)

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

- the penalty increases with the number d of predictors and the variance σ^2 of the irreducible error
- this accounts for the possibility of overtraining, which increases with the complexity of the model

Intuition

- C_p quantifies the in-sample error
- the test error when resampling the training data set
- if $\hat{\sigma}^2$ is an unbiased estimate of σ^2 then C_p is an unbiased estimate of the in-sample error



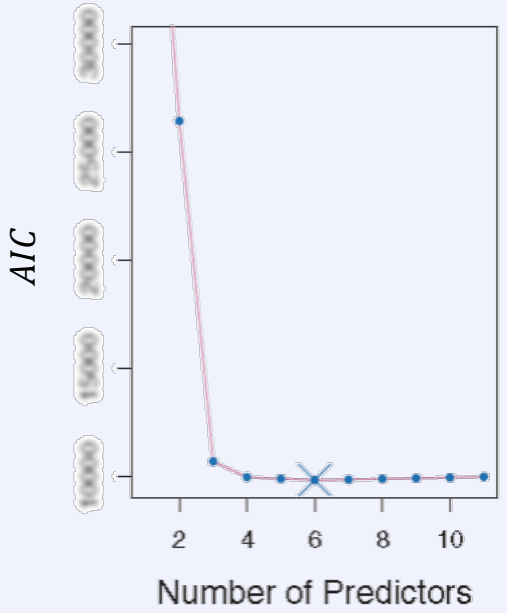
C_p statistic on the Credit data set
The best model involves the variables
income, limit, rating, cards, age, student

Choosing the Optimal Model

5. Akaike's Information Criterion (AIC)

$$AIC = -\frac{1}{n}\log \ell + \frac{k}{n}$$

- for least-square models with Gaussian errors, the maximum likelihood and least-square approaches are equivalent
- that is, we can write the log-likelihood term in terms of RSS



AIC statistic on the Credit data set
The best model involves the variables
income, limit, rating, cards, age, student

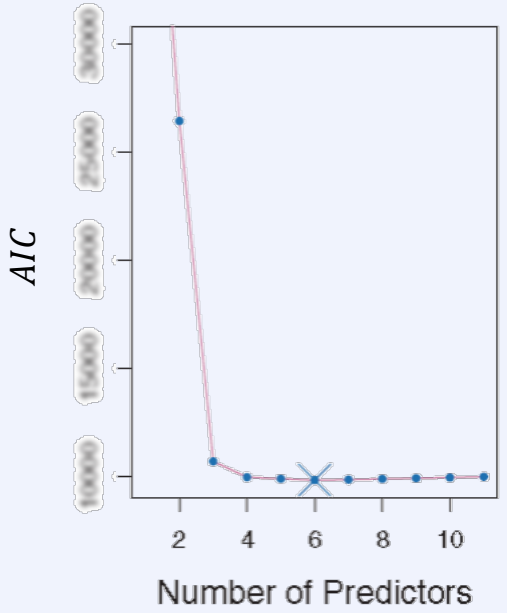
(usually we don't bother dividing by n and just consider $AIC = \log \ell + k$)

Choosing the Optimal Model

5. Akaike's Information Criterion (AIC)

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

- for least-square models with Gaussian errors, maximum likelihood and least-square approaches are equivalent
- the log-likelihood is now re-written in terms of RSS
- there is an additive constant in AIC that we can omit because it does not influence the minimization
- AIC is proportional to C_p and thus yields the same curve



AIC statistic on the Credit data set
The best model involves the variables
income, limit, rating, cards, age, student

Choosing the Optimal Model

Why is least-square with Gaussian error equivalent to max likelihood?

- least-squares with additive Gaussian error

$$Y = f_{\theta(X)} + \epsilon, \qquad \epsilon \sim N(0, \sigma^2)$$

- maximum likelihood

$$\Pr(Y \mid X, \theta) \sim N(f_\theta(X), \sigma^2)$$

$$N(f_\theta(X), \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(Y - f_\theta(X)\right)^2}{2\sigma^2}\right)$$

- log-likelihood

$$\ell(\theta) = -\frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - f_\theta(x_i)\right)^2$$

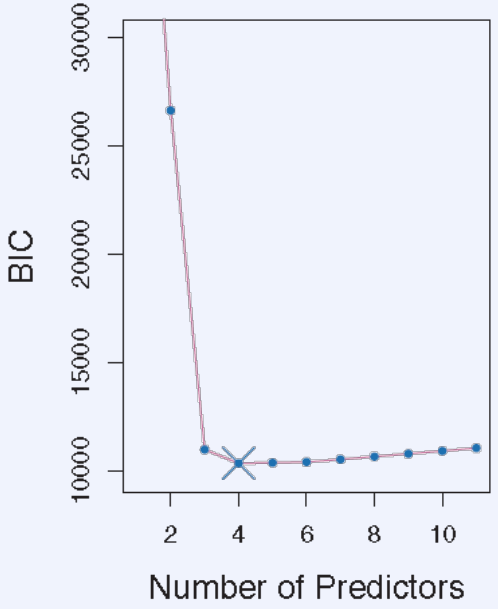
proportional to RSS

Choosing the Optimal Model

5. Bayesian Information Criterion

$$BIC = \frac{1}{n}\log \ell + \frac{k}{2n}\log n$$

- is derived from a Bayesian background and places a heavier penalty on complex models for which $\log(n) > 2$, i.e. $n > 7$



BIC statistic on the Credit data set
The best model involves the variables
income, limit, cards, student

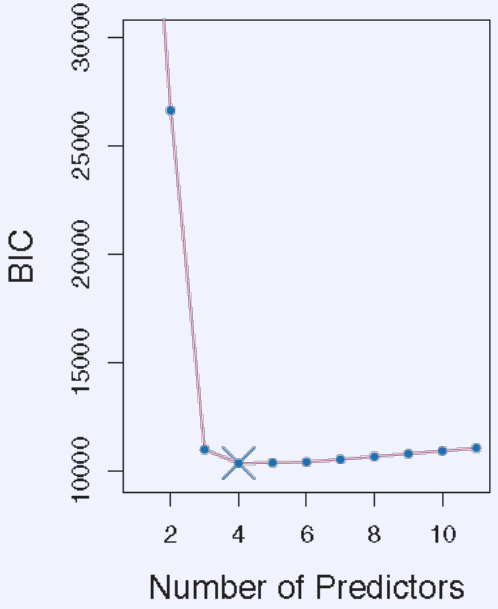
(just like for AIC, we usually don't bother to divide by n, and rather define $BIC = \log \ell + \frac{k}{2}\log n$, ESL 7.7)

Choosing the Optimal Model

6. Bayesian Information Criterion

$$BIC = \frac{1}{n}(RSS + \log(n)\, d\hat{\sigma}^2)$$

- is derived from a Bayesian background and places a heavier penalty on complex models for which $\log(n) > 2$, i.e. $n > 7$
- just like for AIC, we can rewrite the log-likelihood in terms of RSS



BIC statistic on the Credit data set
The best model involves the variables
income, limit, cards, student

Shrinkage Methods

shrinking coefficients rather than setting them to zero

Ridge Regression

Ridge regression

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

- the tuning parameter λ adjusts the relative weight of fit and penalty
- penalizes models that are complex in terms of having large coefficients
- we do not penalize the intercept, so if we center the inputs $x_{ij} \to x_{ij} - \bar{x}_j, \;\; i = 1, \ldots, N$, the intercept is simply

$$\hat{\beta}_0^R = \bar{y} = \sum_{i=1}^{N} \frac{y_i}{N}$$

- the coefficients are then computed as

$$\text{minimize } (\boldsymbol{y} - \boldsymbol{X}\beta)^T (\boldsymbol{y} - \boldsymbol{X}\beta) + \lambda \beta^T \beta$$
$$\hat{\beta}^R = (\boldsymbol{X}^T \boldsymbol{X} + \lambda I)^{-1} \mathbf{X}^T \boldsymbol{y}$$

always nonsingular

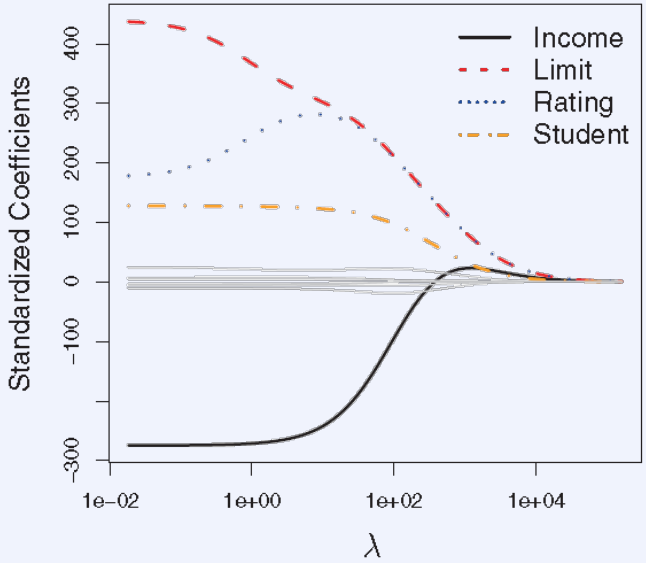
original motivation for ridge regression

- $\lambda = 0$ yields the full linear model, and when $\lambda \to \infty$ we approach the intercept-only model
- selection of λ is critical, done by assessing test error, e.g. with CV

Ridge Regression

Application to the Credit data

- largest coefficients for **income**, **limit**, **rating**, and **student**
- as λ grows, all coefficients are driven to zero
- intermittently, individual coefficients can increase



(ISLR 6.2.1)

Ridge Regression

Application to the Credit data

- largest coefficients for **income**, **limit**, **rating**, and **student**
- as λ grows, all coefficients are driven to zero
- intermittently, individual coefficients can increase

Standard least-square coefficients are scale-equivariant

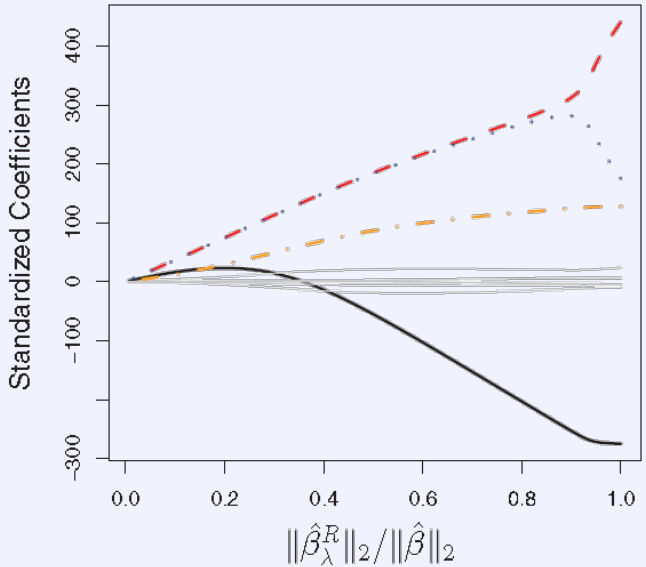
- scaling the inputs with a factor of c leads to scaling the coefficients by $1/c$

Ridge regression coefficients are not scale-equivariant

- always standardize inputs to $\sigma = 1$ before doing ridge regression

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

Sample estimate of the standard deviation of the j^{th} predictor



Model complexity is quantified in terms of the ratio of the L_2 norms of the shrunken and full linear models

Calculating the Ridge Estimates

- if inputs are orthonormal the ridge estimates are scaled versions of least-square estimates, $\hat{\beta}^R = \frac{\hat{\beta}}{1+\lambda}$
- a very plausible quantity for the dimensionality of a model is its effective degrees of freedom (dof)

$$\mathrm{df}(\lambda) = \mathrm{tr}\left[\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\right]$$
$$\mathrm{df}(\lambda) = p \quad \text{if } \lambda = 0$$
$$\mathrm{df}(\lambda) \to 0 \quad \text{for } \lambda \to \infty$$

- the trace $\mathrm{tr}(\mathbf{A})$ of matrix $\mathbf{A}$ is the sum over its diagonal entries

If we have a singular value decomposition of $\mathbf{X}$, i.e. $\mathbf{X} = \mathbf{UDV}$

- $\mathbf{U}$ is a $n \times p$ orthogonal matrix, $\mathbf{D}$ is a $p \times p$ diagonal matrix, and $\mathbf{V}$ is a $p \times p$ orthogonal matrix
- the diagonal entries $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$ of $\mathbf{D}$ are the singular values
- the least squares fitted vector is $\mathbf{X}\hat{\beta}^{\mathrm{ls}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{U}\mathbf{U}^T\mathbf{y}$
- the ridge regression fit is $\mathbf{X}\hat{\beta}^{\mathrm{ridge}} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{UD}(\mathbf{D}^2 + \lambda\mathrm{I})^{-1}\mathbf{D}\mathbf{U}^T\mathbf{y} = \sum_{j=1}^{p} \mathbf{u}_j \frac{d_j^2}{d_j^2+\lambda}\mathbf{u}_j^T\mathbf{y}$
- the dof then takes the form $\mathrm{df}(\lambda) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2+\lambda}$

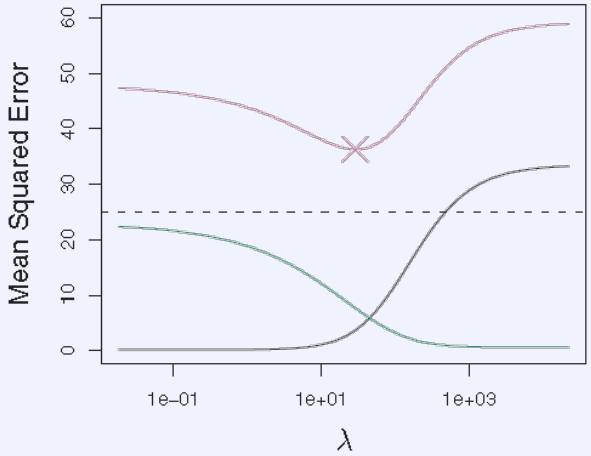
Ridge Regression

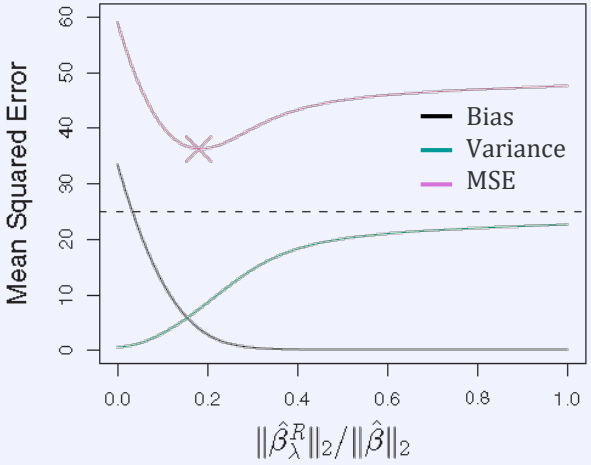
Why does ridge regression improve the full linear model?

- it exploits the bias-variance tradeoff (!)
- especially effective when $p \approx n$

Example Simulated data

- $p = 45, n = 50$
- all inputs related to the response
- if $p > n$ the least-square estimates are not unique, but ridge regression does provide a unique solution
- ridge-regression is also faster than subset selection





(ISLR 6.2.1)

The Lasso

Ridge regression results in dense models

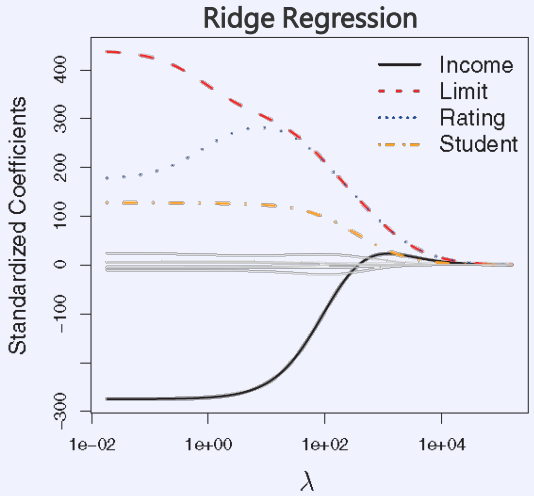
- it does not truly prune features unless $\lambda = \infty$
- many non-zero coefficients limits interpretability of the model

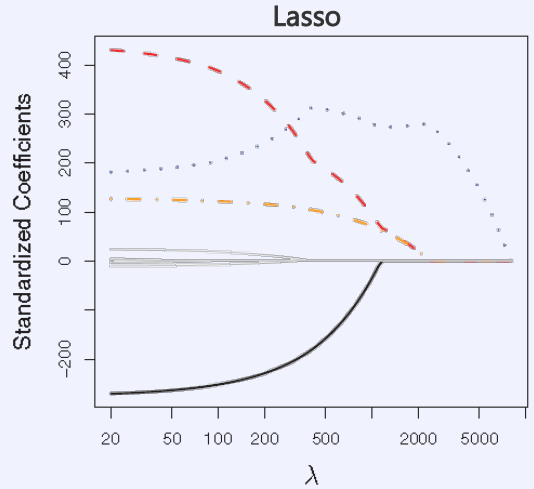
The Lasso

- short for the Least Absolute Shrinkage and Selection Operator

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

- penalizes using the $\mathbf{L}_1$-norm instead of the $\mathbf{L}_2$-norm (ridge)
- yields naturally sparse models, but sensitive to collinearity
- more compute-intensive as it requires solving a quadratic problem
- variants exist that use only a sequence of linear regressions (ESL 3.8)

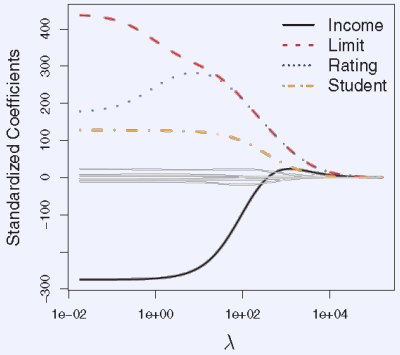


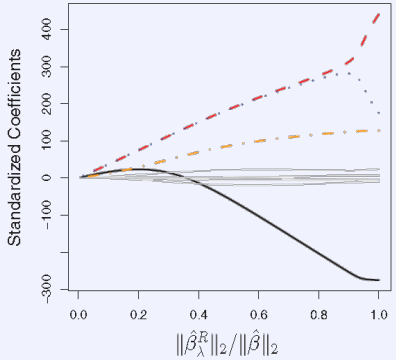


(ISLR 6.2.2)

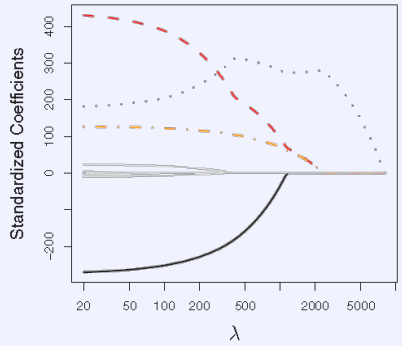
Example Ridge and Lasso

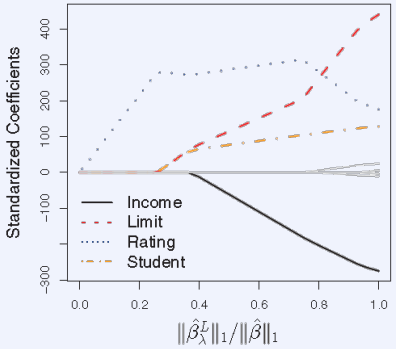
Ridge Regression





Lasso





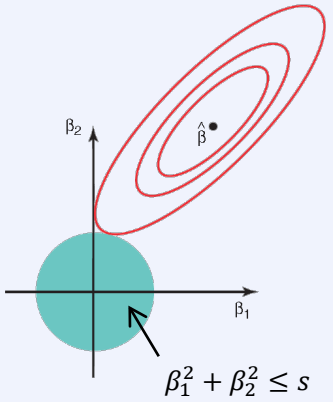
(ISLR 6.2.2)

Intuition Ridge and Lasso

- Ridge Regression

$$\text{minimize } \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \qquad \text{such that } \sum_{j=1}^{p}\beta_j^2 \leq s$$

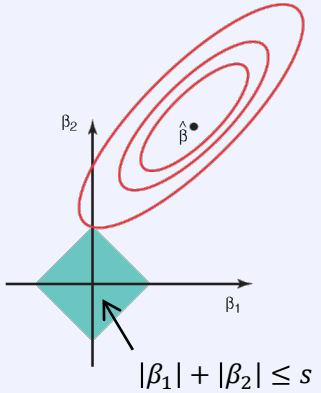
- objective defines a circle in coefficient space
- this generalizes to more dimensions



- Lasso

$$\text{minimize } \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \qquad \text{such that } \sum_{j=1}^{p}|\beta_j| \leq s$$

- objective defines a diamond in coefficient space
- this generalizes to more dimensions



Comparing Ridge and Lasso

Example a simple case

- $n = p$, $\mathbf{X}$ a unit matrix, and we force the intercept to be zero
- residual sum of squares $\sum_{j=1}^{p}(y_j - \beta_j)^2$ is minimized by $\hat{\beta}_j = y_j$
- ridge regression minimizes

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

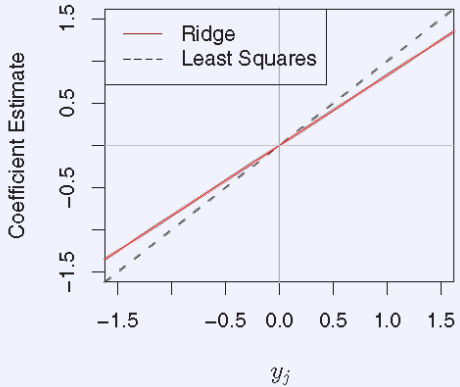
  which yields $\hat{\beta}_j^R = y_j/(1+\lambda)$

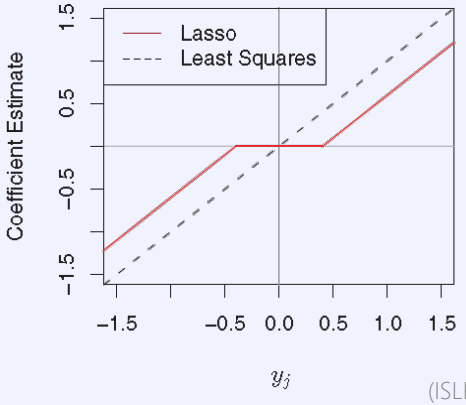
- Lasso minimizes

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

  which yields

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





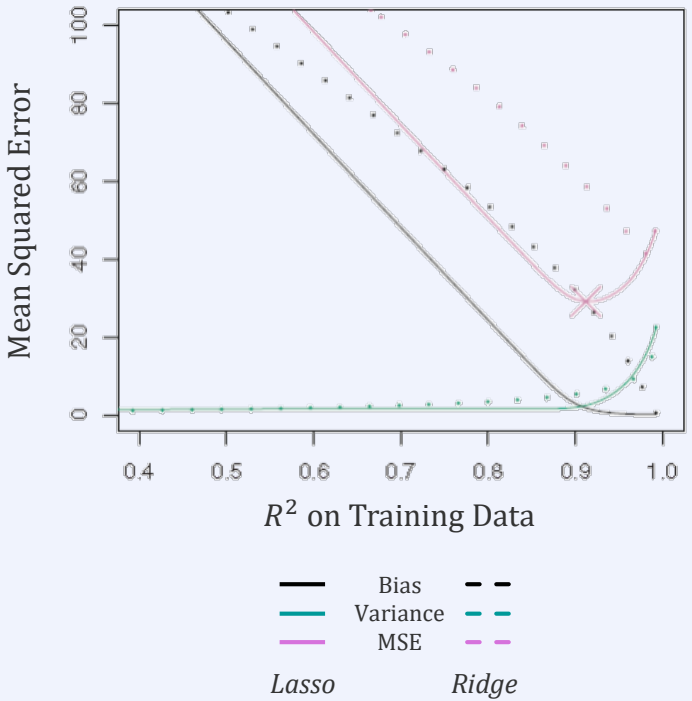
(ISLR 6.2.2)

Comparing Ridge and Lasso

Example Ridge and Lasso

- evaluated in terms of accuracy on simulated data
- $p = 45$
- $n = 50$
- only two inputs related to response

Lasso performs clearly better in this case

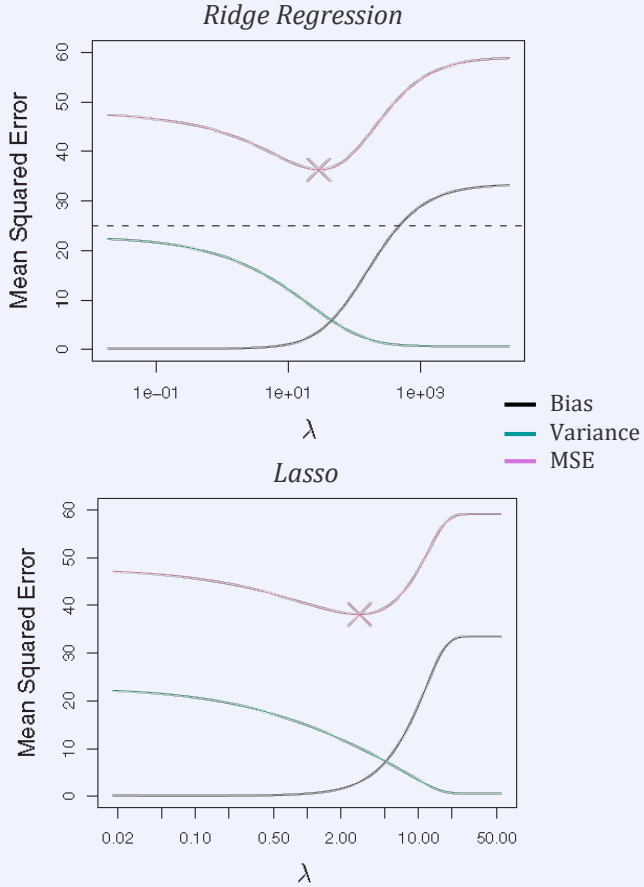


(ISLR 6.2.2)

Comparing Ridge and Lasso

Example Ridge and Lasso

- evaluated in terms of accuracy on simulated data
- $p = 45$
- $n = 50$
- all inputs related to the response



(ISLR 6.2.2)

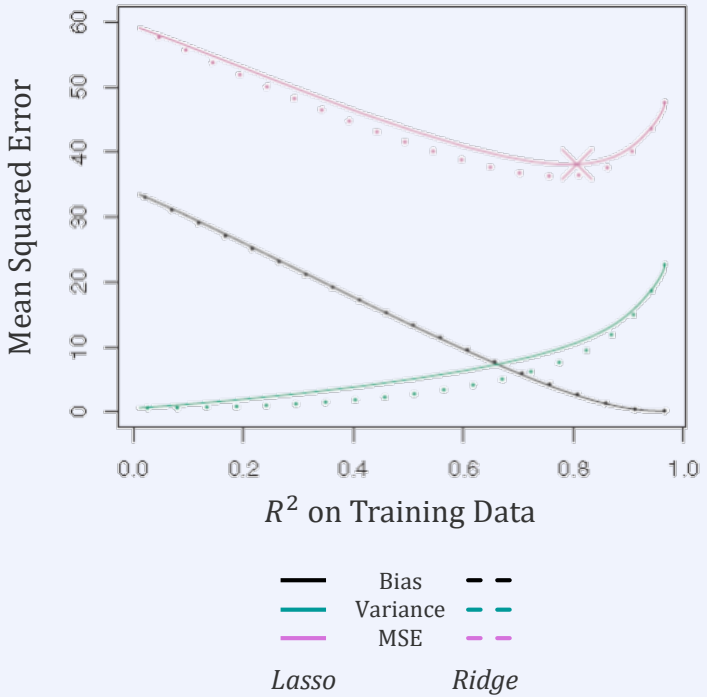
Comparing Ridge and Lasso

Example Ridge and Lasso

- evaluated in terms of accuracy on simulated data
- $p = 45$
- $n = 50$
- all inputs related to the response

Ridge regression is a bit better here

- all true coefficients are nonzero
- Lasso drives some of them to zero



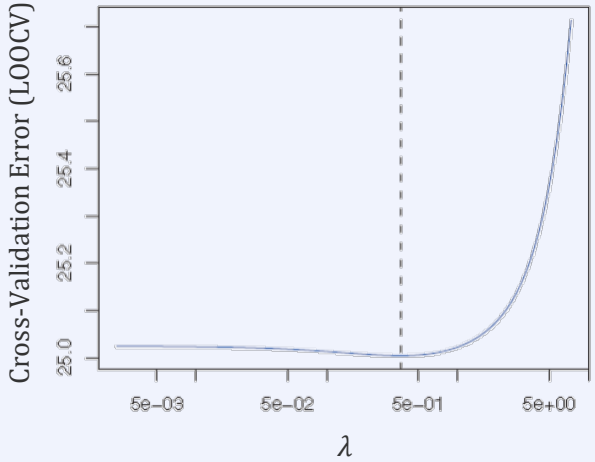
(ISLR 6.2.2)

Selecting the Tuning Parameter

Can be done with cross validation

- select a grid of λ values
- compute cross-validation error for each of the values
- select the value for which the cross-validation error is smallest,
- or, select the largest λ that yields a cross-validation error within one standard deviation of the smallest cross-validation error
- refit the model using all data using that selected value of λ

This is admissible as long as you do not assess test error of the resulting model on any of the training data!



*Model selection on the **Credit** dataset using ridge regression.*

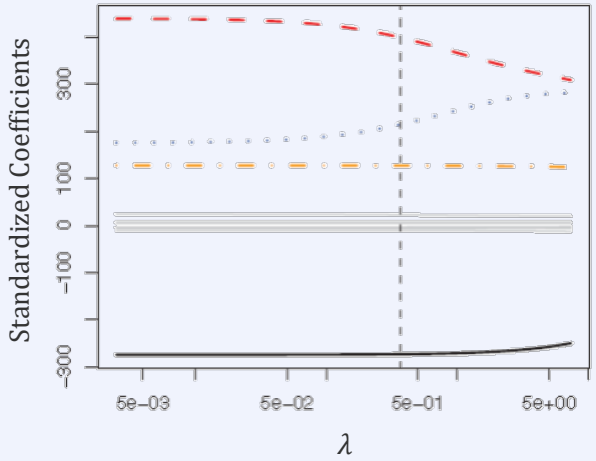
Not much shrinkage is needed

Selecting the Tuning Parameter

Can be done with cross validation

- select a grid of λ values
- compute cross-validation error for each of the values
- select the value for which the cross-validation error is smallest,
- or, select the largest λ that yields a cross-validation error within one standard deviation of the smallest cross-validation error
- refit the model using all data using that selected value of λ

This is admissible as long as you do not assess test error of the resulting model on any of the training data!

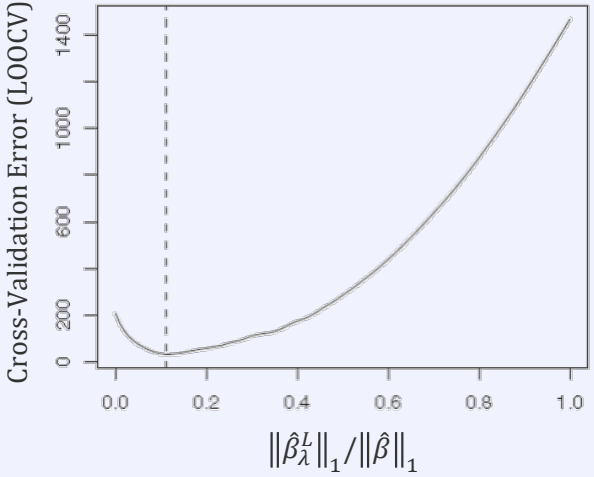


Coefficients as functions of λ on the **Credit** *dataset*

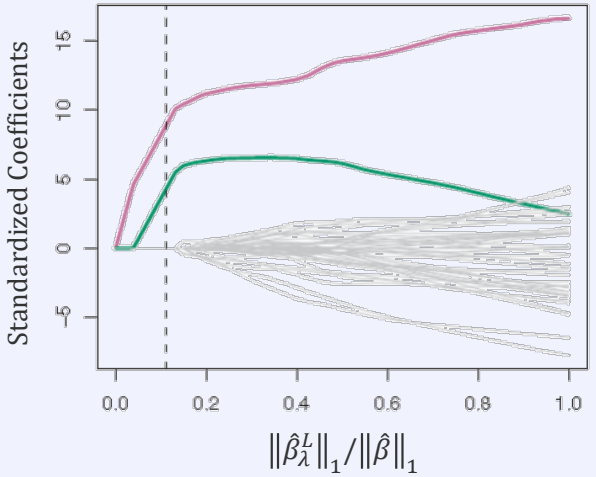
(ISLR 6.2.2)

Selecting the Tuning Parameter

Lasso fit on sparse simulated data set
(only 2 out 45 predictors related to the response)



Here, a lot shrinkage of is needed to weed out unrelated predictors



The full model identifies just 1 predictor

High-Dimensional Data

High-Dimensional Data

Traditionally, data problems tended to be low-dimensional

- far fewer predictors (a handful) than observations (10s up to 1000s)

With new technologies this changed dramatically

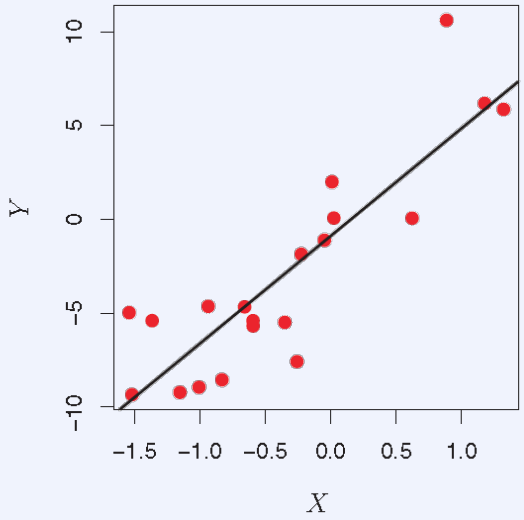
- half a million gene variants (SNPs) for regressing blood pressure measurements on e.g. 200 people
- all the search terms entered by a user in a search engine for marketing purposes

In a high-dimensional problem, the number of features exceeds the number of observations

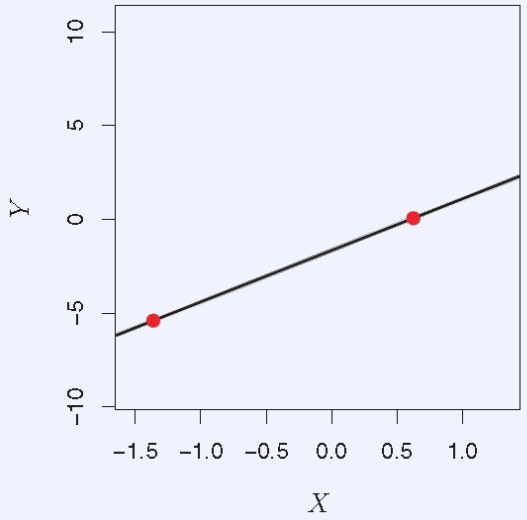
- practically, what we discuss here also applies to cases where $\boldsymbol{p}$ is slightly smaller than $\boldsymbol{n}$

What Goes Wrong in High-Dimensions

In high dimensions, methods like least squares suggest a perfect fit, but are too flexible and overfit



$n = 20, p = 1$

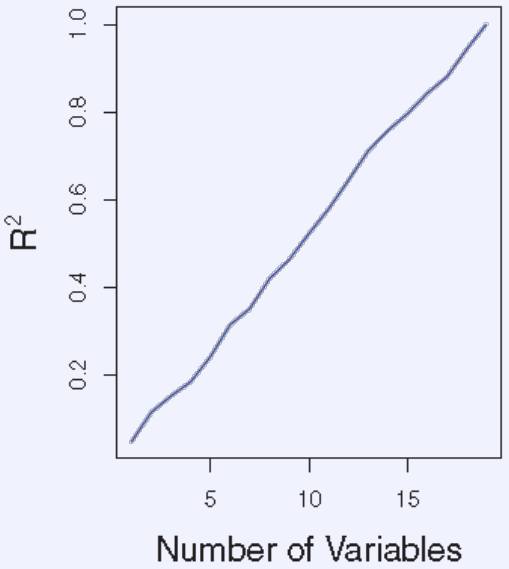


$n = 2, p = 1$

What Goes Wrong in High-Dimensions

Simulated example

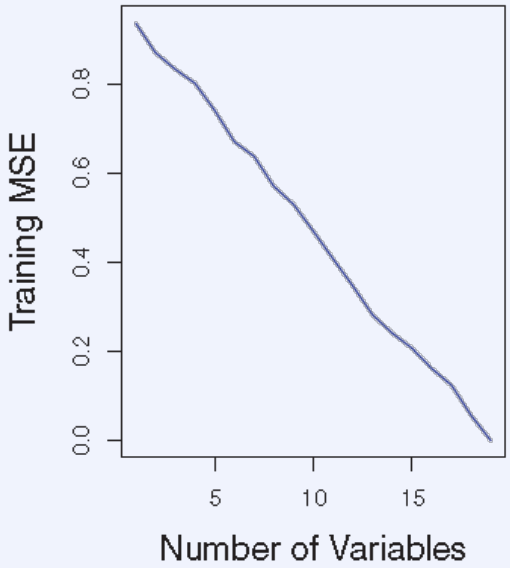
- least-squares regression
- 20 observations
- 1 to 20 features, all completely unrelated to the response
- there is nothing to learn, but nevertheless the correlation rapidly becomes ideal the more features we include



What Goes Wrong in High-Dimensions

Simulated example

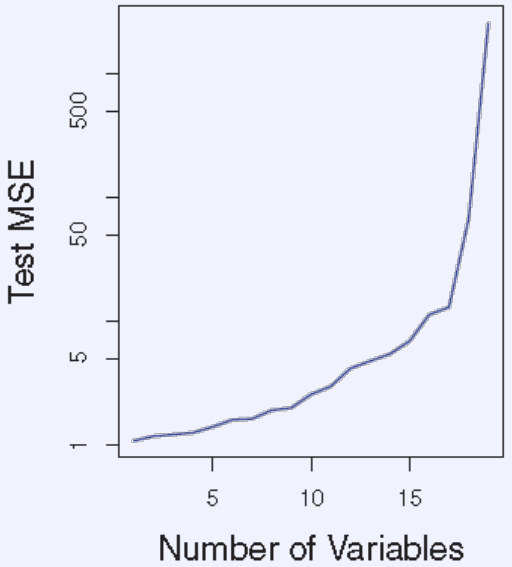
- least-squares regression
- 20 observations
- 1 to 20 features, all completely unrelated to the response
- there is nothing to learn, but nevertheless the correlation rapidly becomes ideal the more features we include
- the training error reduces to zero



What Goes Wrong in High-Dimensions

Simulated example

- least-squares regression
- 20 observations
- 1 to 20 features, all completely unrelated to the response
- there is nothing to learn, but nevertheless the correlation rapidly becomes ideal the more features we include
- the training error reduces to zero
- the test error points very simple models out as the best
- simple model selection techniques like C_p, AIC, BIC do not work well in high-dimensional settings
- adjusted R^2 often approaches 1 and cannot be used either



(ISL 6.4.2)

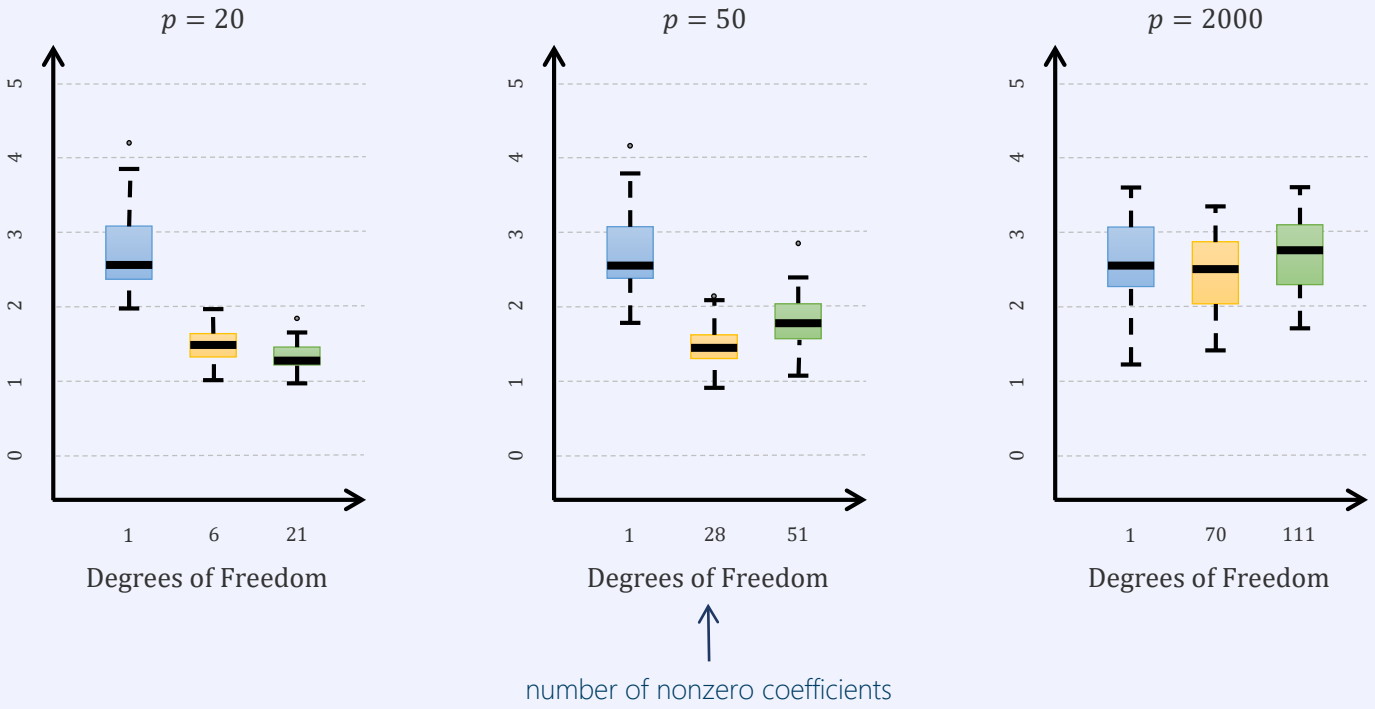
Regression in High Dimensions

Methods for fitting less flexible models are surprisingly suited for high-dimensional data

Simulated example

- Lasso regression
- 100 observations for $p = 20, 50, 2000$
- only 20 features are truly associated with the outcome

Example Regression in High Dimensions

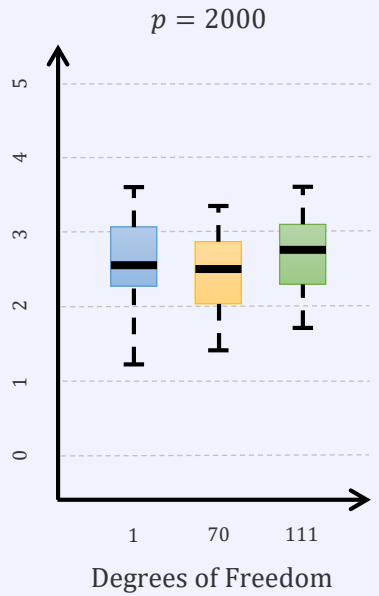


Regression in High Dimensions

Methods for fitting less flexible models are surprisingly suited for high-dimensional data

Simulated example

- Lasso regression
- 100 observations for $p = 20, 50, 2000$
- only 20 features are truly associated with the outcome



Observations

1. regularization can harness problems with high dimensions
2. selecting the appropriate model is crucial
3. test error increases with the number of predictors unrelated to the response (curse of dimensionality)

Interpreting Results in High Dimensions

Multi-co-linearity of predictors is extreme in high dimensions

- any variable is a linear combination of other variables
- we can never know which variables are truly related to the response; we will never find the best coefficients
- all we can do, is find large coefficients for variables that are strongly correlated with those variables that are truly predictive of the response

Example Predicting blood pressure based on 500,000 SNPs

- forward stepwise selection says 17 SNPs provide a predictive model

This does not mean that those SNPs are better than any others at predicting blood pressure

- there will be many sets of 17 SNPs that do the trick
- models on different data sets would be very much different
- so, the model is predictive, but not interpretable

Interpreting Results in High Dimensions

Reporting errors in high-dimensional data fitting

- never use estimates of train error
- never use AIC, BIC or adjusted R^2
- never use p-value statistics based on training data
- instead, use error estimates on independent test sets
  - via RSE or R^2
- or use cross validation