High Dimensional Data Analysis

Chapter 4: High dimensional linear regression

Outline

1. Linear Regression Model
2. High Dimensional Linear Regression

Model Introduction

Assume Y is the response variable, $X_1, \cdots, X_p$ are p covariates (or predictor variables). The relationship between the response variable and the covariates is as follows:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon,$$

where β_0 is the intercept term, $\beta_1, \cdots, \beta_p$ are regression coefficients, and ε is the random model error.
Given $X = x$, the regression function is:

$$g(x) = E(Y|X = x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p,$$

where $X = (X_1, \cdots, X_p)^T$ and $x = (x_1, \cdots, x_p)^T$.

Assume n independent experiments are conducted on $Y, X_1, \cdots, X_p$, obtaining n sets of observations:

$$y_i, x_{i1}, \cdots, x_{ip}, \quad i = 1, \cdots, n$$

Data structure in multiple linear regression:

Response	Covariates				
No.	Y	X_1	X_2	$\cdots$	X_p
1	y_1	x_{11}	x_{12}	$\cdots$	x_{1p}
2	y_2	x_{21}	x_{22}	$\cdots$	x_{2p}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	y_n	x_{n1}	x_{n2}	$\cdots$	x_{np}

The observed sample $\{(y_i, x_i), i = 1, \cdots, n\}$ satisfies:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i, \quad i = 1, \cdots, n.$$

The model errors ε_i satisfy the following Gauss-Markov assumptions:

- $E(\varepsilon_i) = 0, \quad i = 1, \cdots, n;$
- $\text{Var}(\varepsilon_i) = \sigma^2, \quad i = 1, \cdots, n;$
- $\text{Cov}(\varepsilon_i, \varepsilon_j) = 0, \quad i \neq j, \quad i, j = 1, \cdots, n.$

Let

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix},$$

The multiple linear model is written in matrix form as:

$$Y_{n\times 1} = X_{n\times(p+1)}\beta_{(p+1)\times 1} + \epsilon_{n\times 1}.$$

The least squares objective function is:

$$\begin{aligned} Q(\beta) &= \|Y - X\beta\|_2^2 = (Y - X\beta)^T(Y - X\beta) \\ &= Y^TY - 2Y^TX\beta + \beta^TX^TX\beta. \end{aligned}$$

Taking the partial derivative with respect to β and setting it to zero yields the normal equations for β:

$$X^TX\beta = X^TY.$$

The necessary and sufficient condition for the normal equations to have a unique solution is that the rank of X^TX is $p+1$, or the inverse of the matrix X^TX exists.

Least Squares Estimation

Solving the normal equations, the least squares estimate of β is:

$$\hat{\beta} = (X^T X)^{-1} X^T Y.$$

Theorem 1

For the multiple linear regression model, the least squares estimate $\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T Y$ has the following properties:

- $E(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$;
- $\text{Cov}(\hat{\boldsymbol{\beta}}) = \sigma^2 (X^T X)^{-1}$.

With the LS estimate $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_p)^T$, we obtain the following empirical linear regression equation:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p.$$

Estimation of σ^2

From the LS estimate $\hat{\beta}$, the fitted values of Y are:

$$\hat{Y} = X\hat{\beta} = X(X^TX)^{-1}X^TY =: HY.$$

- $\mathbf{H} = X(X^TX)^{-1}X^T$ is called the projection matrix or hat matrix.
- The effective degrees of freedom for the least squares estimate can be defined as:

$$df(ols) = tr(H) = tr(X(X^TX)^{-1}X^T) = tr(X^TX(X^TX)^{-1})$$
$$= tr(I_{p+1}) = p + 1.$$

- The model error vector $\varepsilon = \boldsymbol{Y} - \mathbf{X}\beta$ is an unobservable random vector. Consider the following residual vector:

$$\hat{\varepsilon} = \boldsymbol{Y} - \mathbf{X}\hat{\beta} = \boldsymbol{Y} - \hat{\boldsymbol{Y}} = [\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T]\boldsymbol{Y} = (\mathbf{I}_n - \mathbf{H})\boldsymbol{Y}.$$

- For the ith residual, it can be defined as: $\hat{\varepsilon}_i = y_i - x_i^T\hat{\beta}$, where $x_i = (1, x_{i1}, \cdots, x_{ip})^T$.

- It can be shown that:

$$\mathbf{H}^2 = \mathbf{H}, \quad \mathbf{H}(\mathbf{I}_n - \mathbf{H}) = 0,$$

$$\mathbf{X}^T \hat{\varepsilon} = 0, \quad \hat{\boldsymbol{Y}}^T \hat{\varepsilon} = 0.$$

- The residual sum of squares (RSS) is defined as:

$$RSS = \sum_{i=1}^{n} \hat{\epsilon}_i^2 = \hat{\epsilon}^T \hat{\epsilon}.$$

- The size of RSS reflects the deviation or goodness of fit between the actual data and the theoretical model. A smaller RSS indicates a better fit of the model to the data.

Theorem 2

From the definition of RSS, an unbiased estimate of σ^2 is:

$$\hat{\sigma}^2 = \frac{RSS}{n - p - 1}.$$

Theorem 3

For the multiple linear regression model, further assume the random model error vector $\varepsilon \sim N_n(0, \sigma^2 I_n)$, then:

- $\hat{\beta} \sim N_{p+1}(\beta, \sigma^2 (X^T X)^{-1})$;
- $\frac{RSS}{\sigma^2} = \frac{(n-p-1)\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-p-1}$;
- $\hat{\beta}$ and RSS are mutually independent.

Corollary 1

For the multiple linear regression model, if $\varepsilon \sim N_n(0, \sigma^2 I_n)$, then:

- $\hat{\beta}_k \sim N(\beta_k, \sigma^2 c_{k+1,k+1})$, where $c_{k+1,k+1}$ denotes the $(k+1, k+1)$ element of the matrix $(X^T X)^{-1}$;
- Among all linear unbiased estimates of β_k, $\hat{\beta}_k$ is the unique one with minimum variance, for $k = 0, 1, \cdots, p$.

Confidence interval of β_k

From Corollary 1, the $100(1-\alpha)\%$ confidence interval for β_k is:

$$\hat{\beta}_k \pm t_{n-p-1}\left(\frac{\alpha}{2}\right)\sqrt{\widehat{\mathrm{Var}}(\hat{\beta}_k)}, \quad k = 0, 1, \cdots, p,$$

where

- $\widehat{\mathrm{Var}}(\hat{\beta}_k) = \hat{\sigma}^2 c_{k+1,k+1}$
- $c_{k+1,k+1}$ is the $(k+1, k+1)$ element of the matrix $(X^T X)^{-1}$;
- $t_{n-p-1}(\alpha/2)$ is the upper $\alpha/2$ quantile of the t-distribution with $n-p-1$ degrees of freedom.

Significance Test for Regression Coefficients

- In practical problems, it is often necessary to test whether the kth covariate X_k has a significant effect on the response variable Y, i.e., consider the following hypothesis test:

$$H_{k0} : \beta_k = 0 \Longleftrightarrow H_{k1} : \beta_k \neq 0, \quad k = 1, \cdots, p.$$

- From Corollary 1, Theorem 2, and Theorem 3, it can be shown that when the null hypothesis H_{k0} holds, the statistic

$$T_k = \frac{\hat{\beta}_k}{\hat{\sigma}\sqrt{c_{k+1,k+1}}} \sim t_{n-p-1}, \quad k = 1, \cdots, p.$$

- If $|T_k| \geq t_{n-p-1}(\alpha/2)$ or the p-value $p_k = \mathbb{P}(t_{n-p-1} \geq |T_k|) < \alpha/2$, then reject the null hypothesis H_{k0}, concluding that $\beta_k \neq 0$.

Real Example: Prostate Cancer Dataset `prostate`

Consider the prostate cancer dataset `prostate` from the `faraway` package.
This dataset contains 97 samples and 9 variables:
`lpsa` (log of PSA),
`lcavol` (log of cancer volume),
`lweight` (log of prostate weight),
`age` (patient age),
`lbph` (log of benign prostatic hyperplasia amount),
`svi` (seminal vesicle invasion),
`lcp` (log of capsular penetration),
`gleason` (Gleason score), and
`pgg45` (percentage of Gleason scores 4 or 5).
Consider the following multiple linear regression model:

$$\begin{aligned}\texttt{lpsa} &= \beta_0 + \beta_1\texttt{lcavol} + \beta_2\texttt{lweight} + \beta_3\texttt{age} + \beta_4\texttt{lbph} \\ &\quad + \beta_5\texttt{svi} + \beta_6\texttt{lcp} + \beta_7\texttt{gleason} + \beta_8\texttt{pgg45} + \varepsilon.\end{aligned}$$

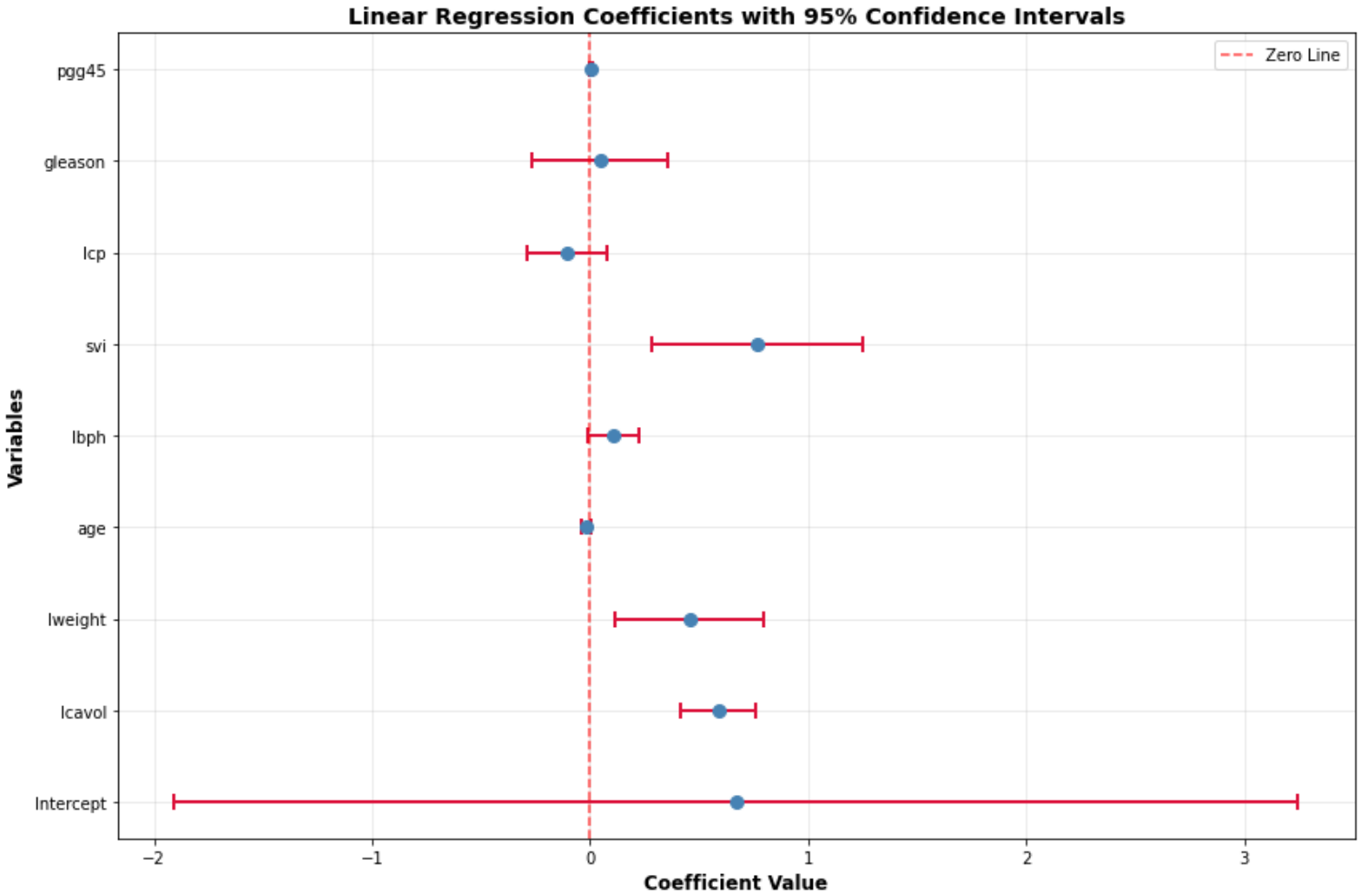
Variable	Coef Estimate	Std. Error	t statistic	p-value
(Intercept) β_0	0.669337	1.296387	0.516	0.60693
`lcavol` β_1	0.587022	0.087920	6.677	2.11e-09 ***
`lweight` β_2	0.454467	0.170012	2.673	0.00896 **
`age` β_3	-0.019637	0.011173	-1.758	0.08229 .
`lbph` β_4	0.107054	0.058449	1.832	0.07040 .
`svi` β_5	0.766157	0.244309	3.136	0.00233 **
`lcp` β_6	-0.105474	0.091013	-1.159	0.24964
`gleason` β_7	0.045142	0.157465	0.287	0.77503
`pgg45` β_8	0.004525	0.004421	1.024	0.30886

$n = 97 \quad R^2 = 0.6548 \quad R^2_{adj} = 0.6234 \quad \text{Residual SE} = 0.7084 \quad \text{d.f.} = 88$

F-statistic: 20.86 on 8 and 88 DF, p-value: $< 2.2e-16$

	2.5 %	97.5 %
(Intercept)	-1.906960983	3.245634379
lcavol	0.412298699	0.761744954
lweight	0.116603435	0.792331414
age	-0.041840618	0.002566267
lbph	-0.009101499	0.223209561
svi	0.280644232	1.251670420
lcp	-0.286344443	0.075395916
gleason	-0.267786053	0.358069248
pgg45	-0.004260932	0.013311395

Real Example



- The empirical regression equation is: $\texttt{lpsa} \approx 0.6693 + 0.5870 \times \texttt{lcavol} + 0.4545 \times \texttt{lweight} - 0.0196 \times \texttt{age} + 0.1071 \times \texttt{lbph} + 0.7662 \times \texttt{svi} - 0.1055 \times \texttt{lcp} + 0.0451 \times \texttt{gleason} + 0.0045 \times \texttt{pgg45}$;
- From the p-values: Variables `lcavol`, `lweight`, and `svi` are statistically significant in the linear regression, while variables age, `lbph`, `lcp`, `gleason`, and pgg45 are not significant.
- $R^2 = 0.6548$ indicates that the empirical regression equation fits the data reasonably well;
- The test for the regression equation: the p-value of the F-statistic is 2.2×10^{-16}, much smaller than the significance level $\alpha = 0.05$, indicating that the empirical regression equation is significant.

Python code for Real Example

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler

# Read data
file_path = "E:/data/prostate.csv"
prostate = pd.read_csv(file_path)
```

Python code for Real Example

```
# 1. Linear regression model
lm_reg = smf.ols('lpsa ~ lcavol + lweight + age + lbph + svi
                  + lcp + gleason + pgg45',
                  data=prostate).fit()

# Print model summary
print(lm_reg.summary())

# 2. Confidence intervals
conf_int = lm_reg.conf_int()
print(conf_int)

# Create results dataframe
conf_int.columns = ['Lower CI', 'Upper CI']
coefficients = lm_reg.params
```

Python code for Real Example

```
results_df = pd.DataFrame({
    'Coefficient': coefficients,
    'Lower_CI': conf_int['Lower CI'],
    'Upper_CI': conf_int['Upper CI']
})

# Plot coefficients with confidence intervals
plt.figure(figsize=(12, 8))
coefficient_names = results_df.index
y_pos = np.arange(len(coefficient_names))

plt.errorbar(results_df['Coefficient'], y_pos,
 xerr=[results_df['Coefficient'] - results_df['Lower_CI'],
       results_df['Upper_CI'] - results_df['Coefficient']],
       fmt='o', markersize=8, capsize=5, capthick=2,
       color='steelblue', ecolor='crimson', linewidth=2)
```

Python code for Real Example

```
plt.axvline(x=0, color='red', linestyle='--', alpha=0.7, label
plt.yticks(y_pos, coefficient_names)
plt.xlabel('Coefficient Value', fontsize=12, fontweight='bold'
plt.ylabel('Variables', fontsize=12, fontweight='bold')
plt.title('Linear Regression Coefficients with 95% Confidence
          fontsize=14, fontweight='bold')
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()
```

High Dimensional Linear Regression

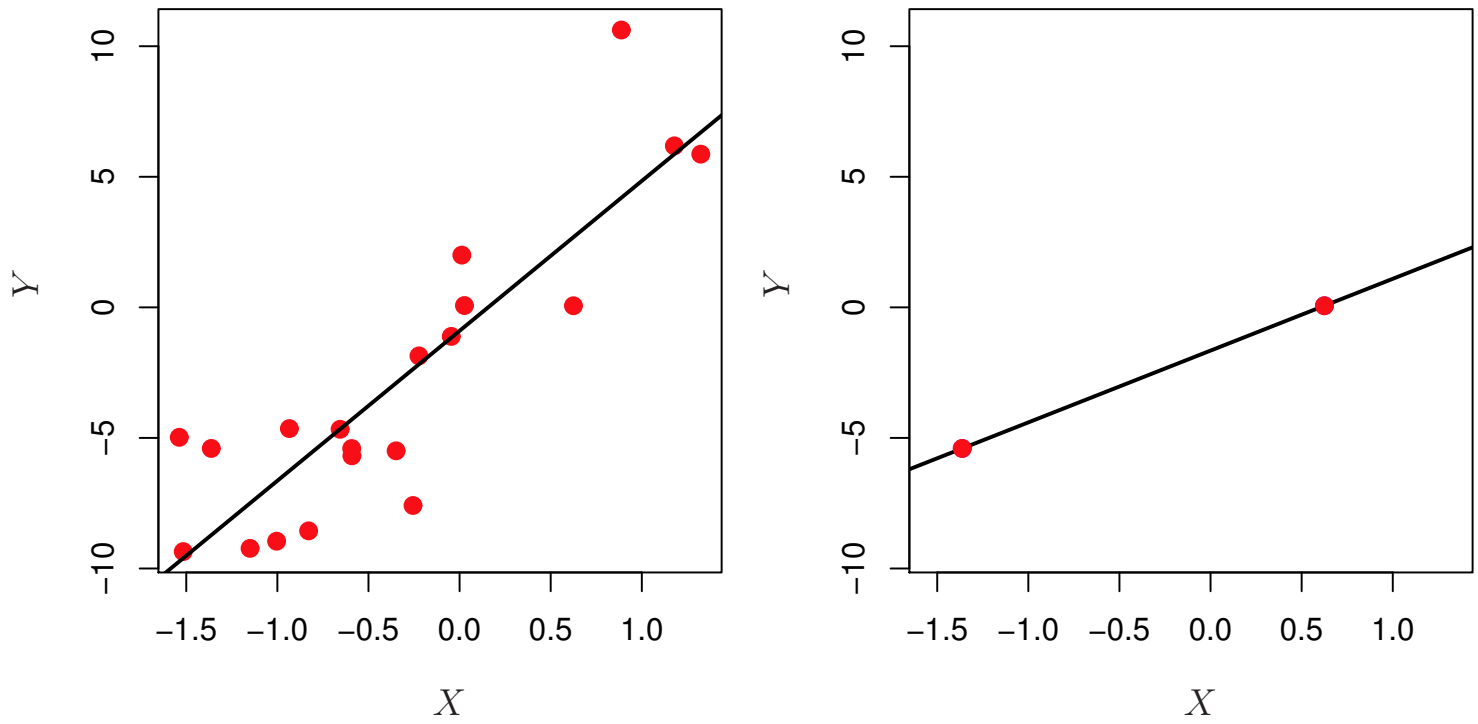
- In statistics and machine learning, we aim to explain output $Y \in \mathcal{Y}$ from input $X \in \mathcal{X} \subset \mathbb{R}^p$.
- We observe a dataset $D_n = \{(X_i, Y_i)\}_{1 \leq i \leq n}$ of i.i.d. observations.
- Today's focus: high-dimensional input spaces where $p \gg n$.
- Two main motivations:
  - **Prediction accuracy**: Classical models fail when $p \gg n$.
  - **Model interpretability**: Remove irrelevant features for better understanding.

Why High-Dimensional Data?

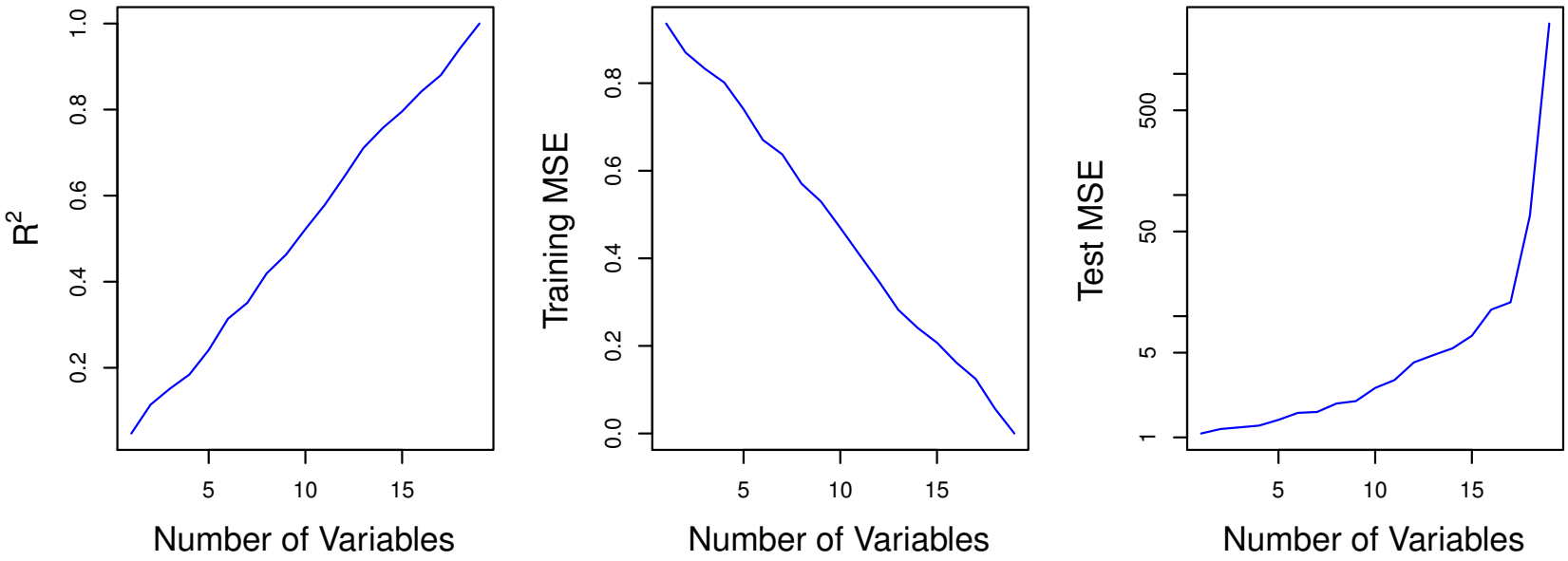
- Exponential growth in data volume (IBM: 10^{18} bytes created daily according to IBM six years ago, and 90% of data is less than two years old).
- Modern applications record thousands to millions of features:
  - DNA sequencing and genomics
  - Image and video analysis
  - Consumer preference tracking (cookies)
  - Astrophysics and sensor networks

The Curse of Dimensionality

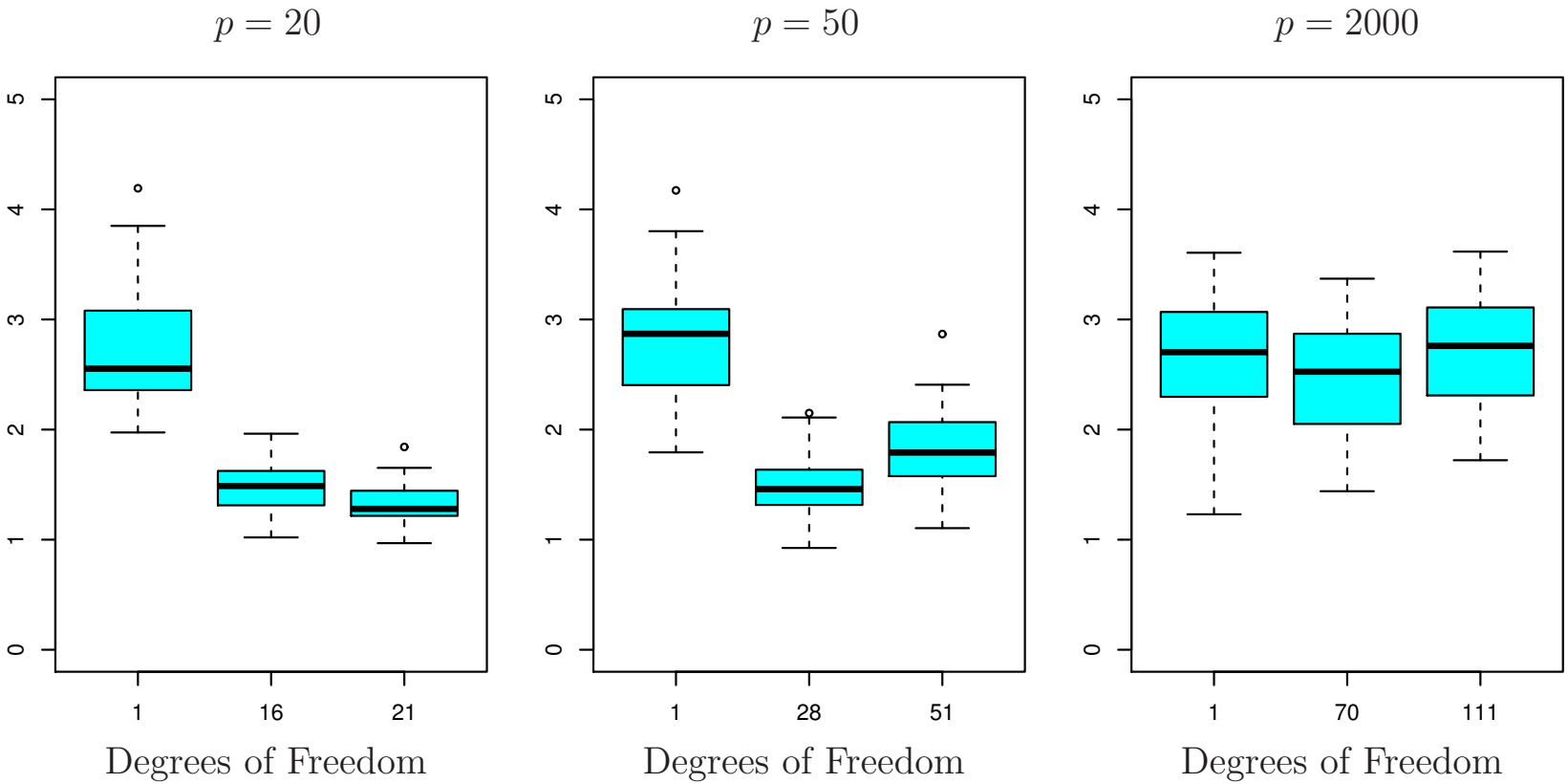
- High-dimensional spaces are vast, data points are isolated
- Accumulation of small errors in many directions produces large global error
- Events that are rare in low dimensions may be common in high dimensions



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).



On a simulated example with n = 20 training observations, features that are completely unrelated to the outcome are added to the model. **Left:** The R2 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.



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 λ. For ease of interpretation, rather than reporting λ, 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.

Problem 1: Coverage of Space

- Input space: $\mathcal{X} = [0, 1]^p$
- Number of points needed to cover space at radius ε in L2 norm is of order $\sim 1/\varepsilon^p$ (exponential in p)
- Consequence: In high dimensions, no point in dataset is likely to be close to a new input
- This challenges methods like K-Nearest Neighbors

Problem 2: Meaningless Distances

- Let $X, X' \sim \text{Uniform}([0,1]^p)$
- Expected squared L2-distance:

$$\begin{aligned}\mathbb{E}\left[\|X - X'\|^2\right] &= \sum_{i=1}^{p} \mathbb{E}[(X_i - X_i')^2] = p\mathbb{E}[(X_1 - X_1')^2] \\ &= p \int_0^1 \int_0^1 (x - x')^2 dx dx' = \frac{p}{6}\end{aligned}$$

- Standard deviation:

$$\begin{aligned}\sqrt{\text{Var}\left[\|X - X'\|^2\right]} &= \sqrt{\sum_{i=1}^{p} \text{Var}[(X_i - X_i')^2]} \\ &= \sqrt{p\text{Var}[(X_1 - X_1')^2]} = \frac{\sqrt{7p}}{6\sqrt{5}} \sim 0.2\sqrt{p}\end{aligned}$$

- Consequence: All points tend to be at similar distances from one another
- Nearest neighbor concept becomes less meaningful

Problem 3: Instability of OLS

- Linear model: $Y_i = x_i^\top \beta^* + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$
- OLS estimator: $\widehat{\beta} = (X^\top X)^{-1} X^\top Y$ satisfies

$$\mathbb{E}\left[\|\widehat{\beta} - \beta^*\|^2\right] = \mathsf{Tr}\Big((X^T X)^{-1}\Big)\sigma^2$$

- Under orthogonal design: $\mathbb{E}\left[\|\widehat{\beta} - \beta^*\|^2\right] = p\sigma^2$
- Variance increases linearly with dimension
- OLS only works for $p \leq n$ (otherwise $X^\top X$ not invertible)
- Need for regularization in high dimensions

Object: Intrinsic Low Complexity

- Despite curse of dimensionality, data often has intrinsic low complexity:
  - Sparsity: Only a few features are relevant
  - Low-dimensional structure: Data lies on or near a low-dimensional manifold
- This structure can come from:
  - The data itself (e.g., images have local correlations)
  - The learning method (e.g., kernel methods implicitly use low-dimensional representations)
- Exploiting this structure is key to success in high dimensions

Approaches to High-Dimensional Data

1. **Subset Selection**: Identify a subset of $s \ll p$ predictors
   - Fit model (e.g., OLS) on selected variables only
2. **Regularization**: Add constraints or penalties to estimation
   - Ridge regression (ℓ_2 penalty)
   - Lasso (ℓ_1 penalty)
   - ...
3. **Dimension Reduction**: Project data to lower-dimensional space. If we consider linear transformation, we may project the p predictors into a s-dimensional space with $s \ll p$. This is achieved by computing s different linear combination or projections of the variables. Then these projections are used as new features to fit a simple model for instance by least squares.
   - Principal Component Analysis (PCA)
   - Partial Least Squares (PLS)
   - ...

Regularization

- Larger models $m \supset m'$ will always get smaller empirical risk because of overfitting.
- Penalty needed to avoid overfitting (larger models always have smaller empirical risk).
- Select best model using penalized criterion:

$$\widehat{m} = \arg \min_{m \in \mathcal{M}} \left\{ \| Y - X\widehat{\beta}_m \|^2 + \text{penalty}(m) \right\}$$

Regularization

ℓ_2-norm–Ridge Regression

- Penalization: penalty$(m) = \lambda\|\beta\|_2^2$, that is

$$\widehat{\beta}_\lambda \in \arg\min_{\beta\in\mathbb{R}^p} \left\{\|Y - X\beta\|^2 + \lambda\|\beta\|_2^2\right\}$$

  where $\lambda > 0$ is the tuning parameter, to be determined separately.

- $$\widehat{\beta}_\lambda = (X^T X + \lambda I_p)^{-1} X^T Y$$

  is the ridge estimator given the tuning parameter λ.
- Ridge regression does have one obvious disadvantage: unlike subset selection, which will generally select models that involve just a subset of the variables, ridge regression will include all p predictors in the final model.

Credit Data Example

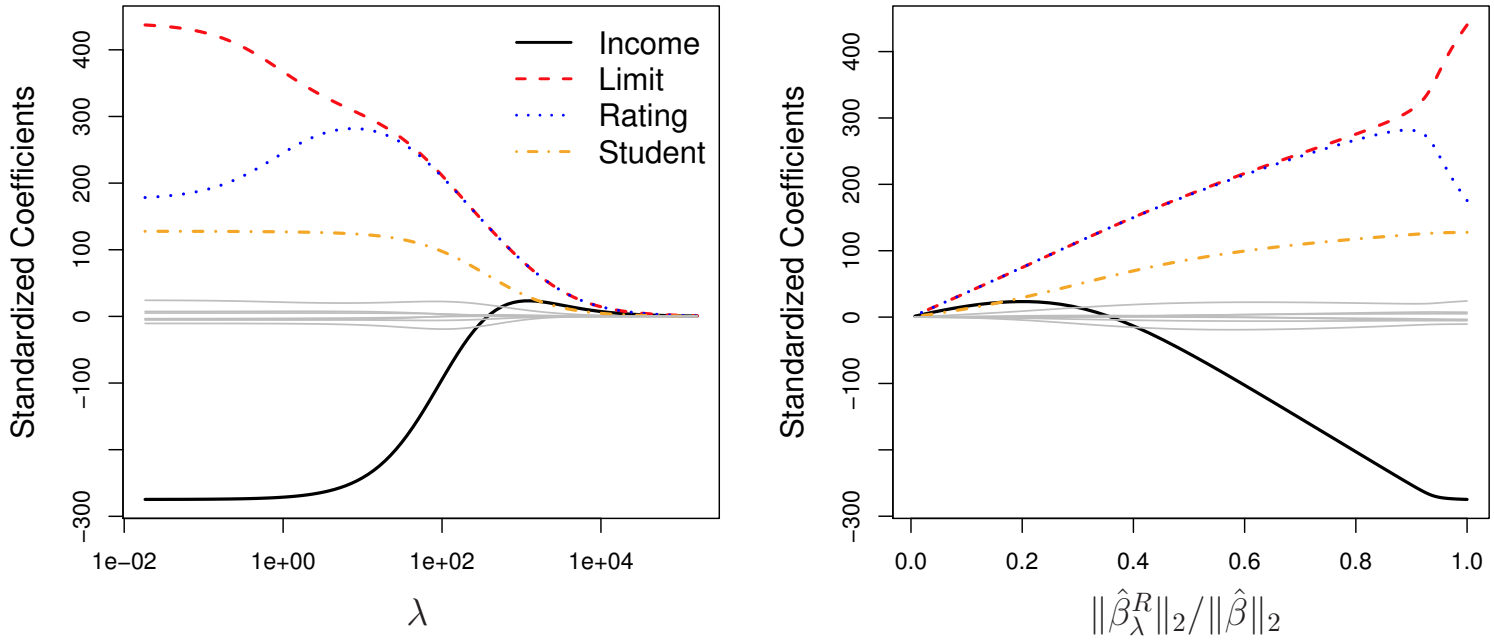


Figure: Standardized Coefficients for different regularization parameters

Details of Previous Figure

- In the left-hand panel, each curve corresponds to the ridge regression coefficient estimate for one of the ten variables, plotted as a function of λ
- The right-hand panel displays the same ridge coefficient estimates as the left-hand panel, but instead of displaying λ on the x-axis, we now display $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$, where $\hat{\beta}$ denotes the vector of least squares coefficient estimates
- The notation $\|\beta\|_2$ denotes the ℓ_2 norm of a vector: $\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$

Why Does Ridge Regression Improve Over Least Squares?

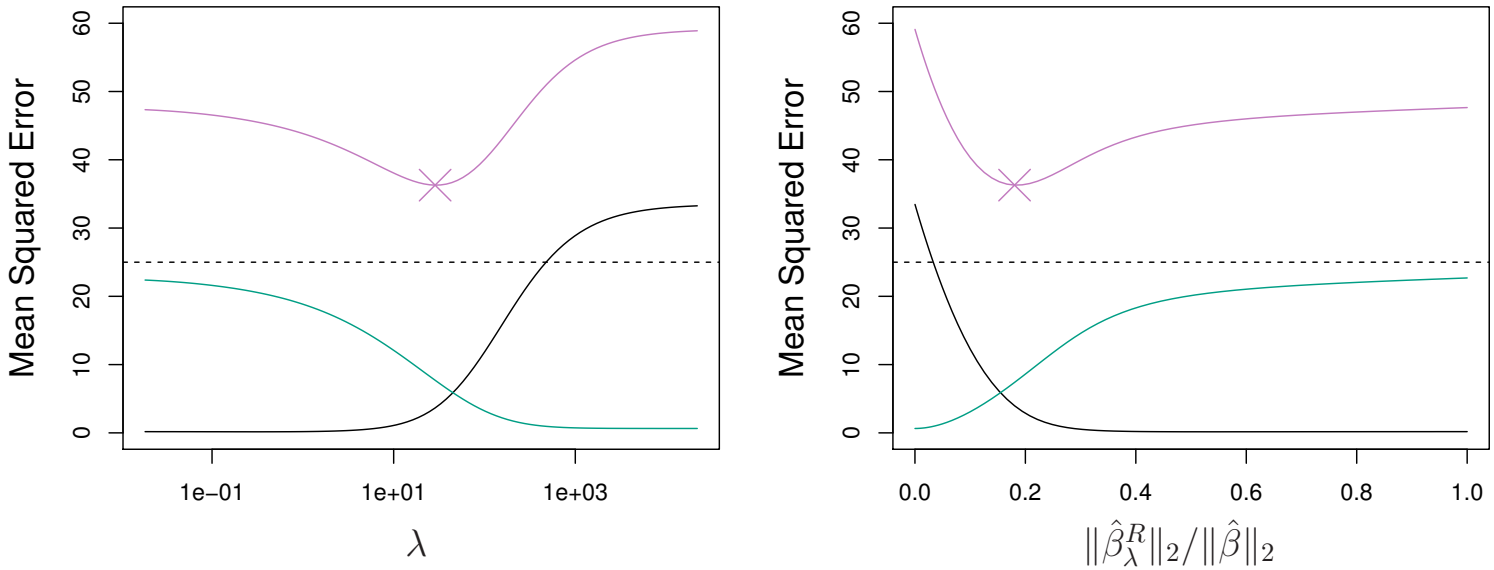


Figure: Bias-Variance tradeoff for Ridge Regression

Simulated data with $n = 50$ observations, $p = 45$ predictors, all having nonzero coefficients. Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions.

Regularization

ℓ_1-norm–Lasso Regression

- Replace ℓ_2-norm with ℓ_1-norm: penalty$(m) = \lambda\|\beta\|_1$, that is

$$\widehat{\beta}_\lambda \in \arg\min_{\beta\in\mathbb{R}^p} \left\{\|Y - X\beta\|^2 + \lambda\|\beta\|_1\right\}$$

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero.
- However, in the case of the lasso, the ℓ_1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large.
- Hence, much like best subset selection, the lasso performs variable selection. That is, the lasso yields sparse models, models that involve only a subset of the variables.
- As in ridge regression, selecting a good value of λ for the lasso is critical; cross-validation is again the method of choice.

Example: Credit Dataset

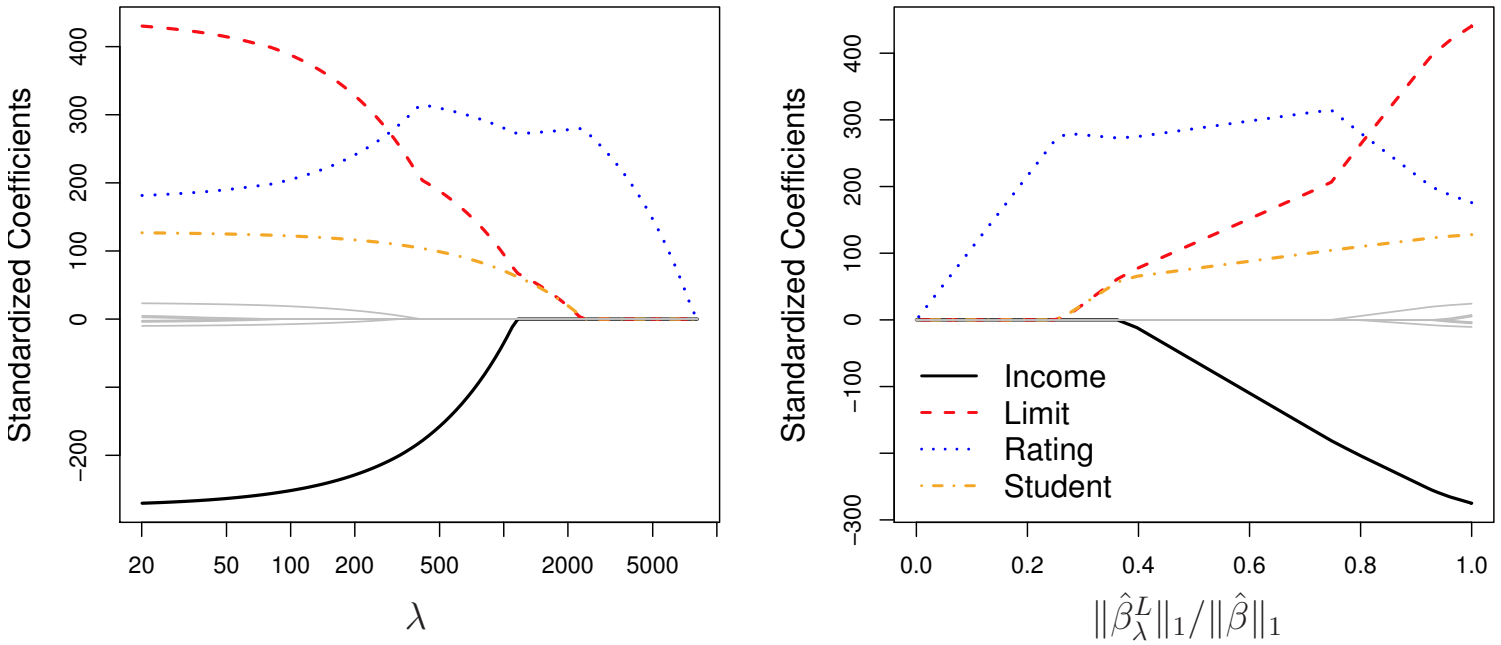
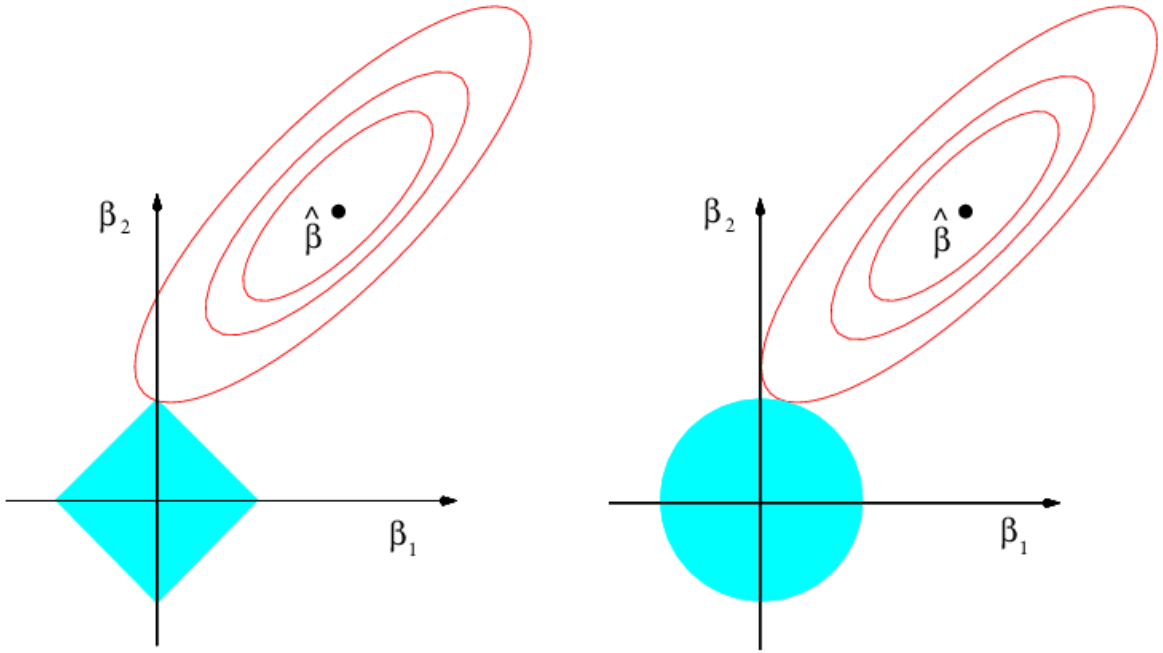


Figure: Lasso Coefficient Paths

Regularization

Geometric Insight



- Left: ℓ_1 regularization produces sparse solutions (some coefficients exactly zero)
- Right: ℓ_2 regularization shrinks coefficients but doesn't set them to zero
- Sharp corners of ℓ_1-ball cause coefficient selection

Comparing the Lasso and Ridge Regression

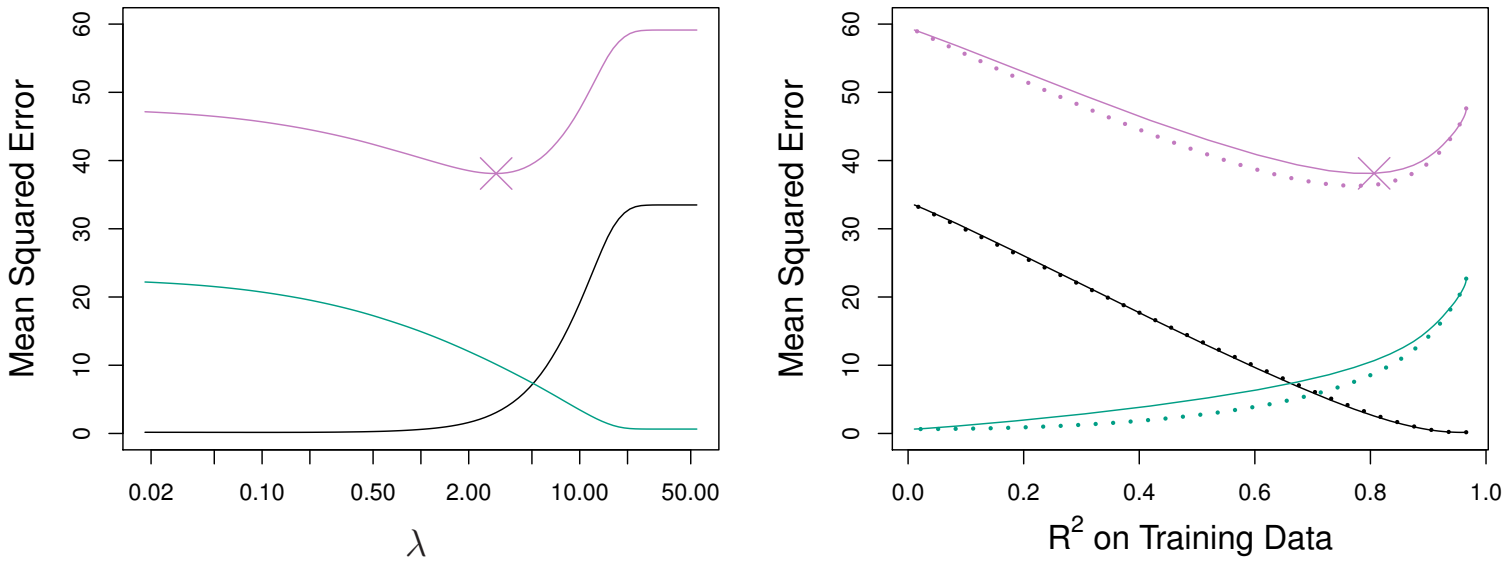


Figure: Comparison of Lasso and Ridge Regression

Left: Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso. **Right:** Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dashed).

Comparing the Lasso and Ridge Regression: continued

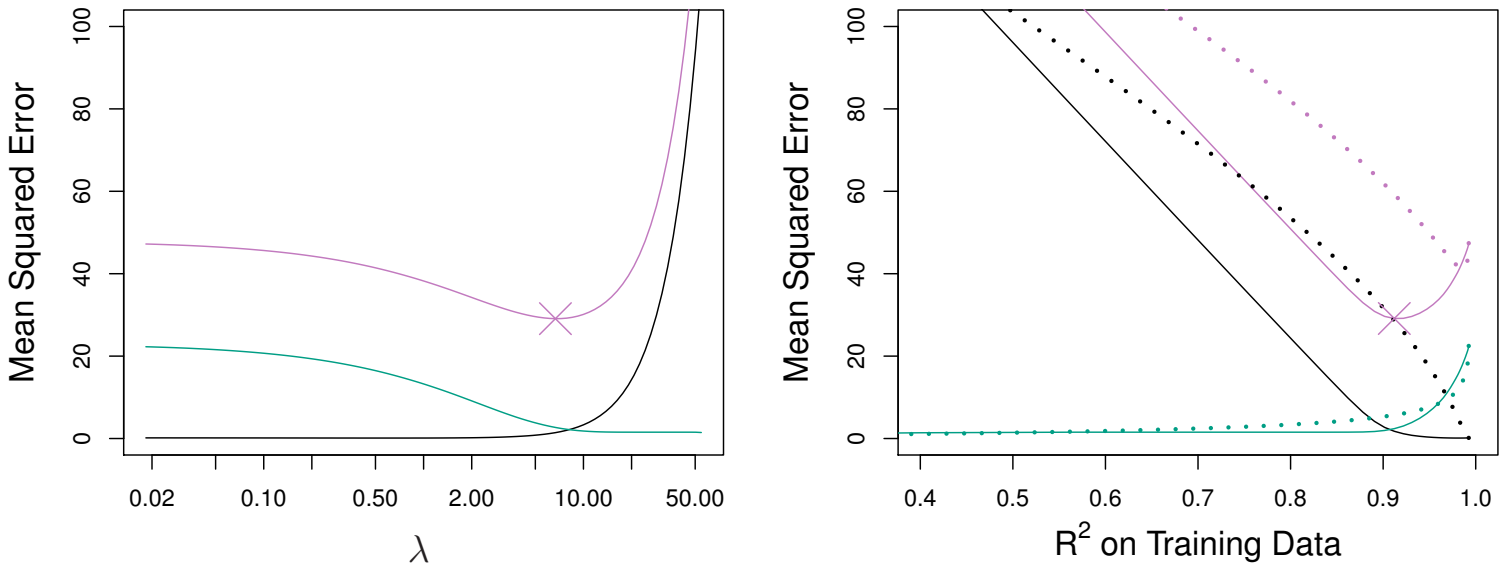


Figure: Comparison of Lasso and Ridge Regression (sparse case)

Left: Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso. The simulated data is similar to previous but with only two relevant predictors. **Right:** Comparison between lasso (solid) and ridge (dashed).

Conclusions

- These two examples illustrate that neither ridge regression nor the lasso will universally dominate the other
- In general, one might expect the lasso to perform better when the response is a function of only a relatively small number of predictors
- However, the number of predictors that is related to the response is never known *a priori* for real data sets
- A technique such as cross-validation can be used to determine which approach is better on a particular data set

Computing the Lasso Estimator

Three main algorithms:

1. **Coordinate descent**:
   - Repeatedly minimize objective with respect to each coordinate
   - Converges due to convexity
2. **FISTA (Fast Iterative Shrinkage-Thresholding Algorithm)**:
   - Uses explicit formula from orthonormal case
   - Accelerated proximal gradient method
3. **LARS (Least Angle Regression)**:
   - Computes full regularization path $\lambda \mapsto \widehat{\beta}_\lambda$
   - Solves sequence of linear equations at breakpoints
   - Slower but provides complete solution path

Regularization

Lasso Variants and Extensions

- **Debiased Lasso**:
  - Lasso is biased due to shrinkage
  - Two-step procedure: Use Lasso for variable selection, then refit OLS on selected variables
- **Unpenalized intercept**:
  - In practice, intercept is often not penalized
  - Solve: $\min \sum (Y_i - \beta_0 - \beta^\top X_i)^2 + \lambda \|\beta\|_1$
- **Group Lasso**:
  - Variables are sparse by groups $G_k \subset \{1, \ldots, p\}$
  - All coefficients in group are either all zero or all non-zero
- **Elastic Net**:
  - Combination of ℓ_1 and ℓ_2 regularization
  - $\min \|Y - X\beta\|^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2$
  - Handles correlated variables better than Lasso

Selecting the Tuning Parameter for Ridge Regression and Lasso

- For ridge regression and lasso we require a method to determine which model is best
- We need to select a value for the tuning parameter λ or the constraint s
- Cross-validation provides a simple way to tackle this problem
- We choose a grid of λ values and compute the cross-validation error rate for each value
- Select the tuning parameter value for which the cross-validation error is smallest
- Finally, refit the model using all observations and the selected tuning parameter

Credit Data Example

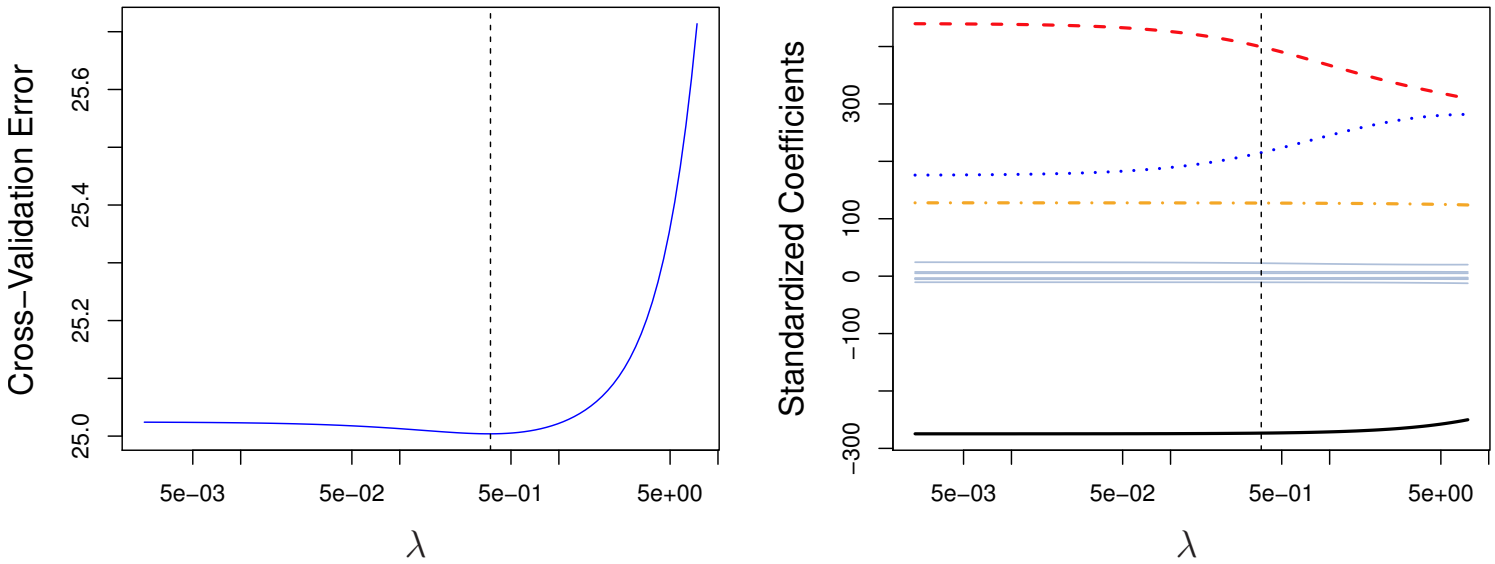


Figure: Ridge Regression Cross-Validation

Left: Cross-validation errors for ridge regression with various λ. **Right:** Coefficient estimates as a function of λ. The vertical dashed lines indicate the value of λ selected by cross-validation.

Simulated Data Example

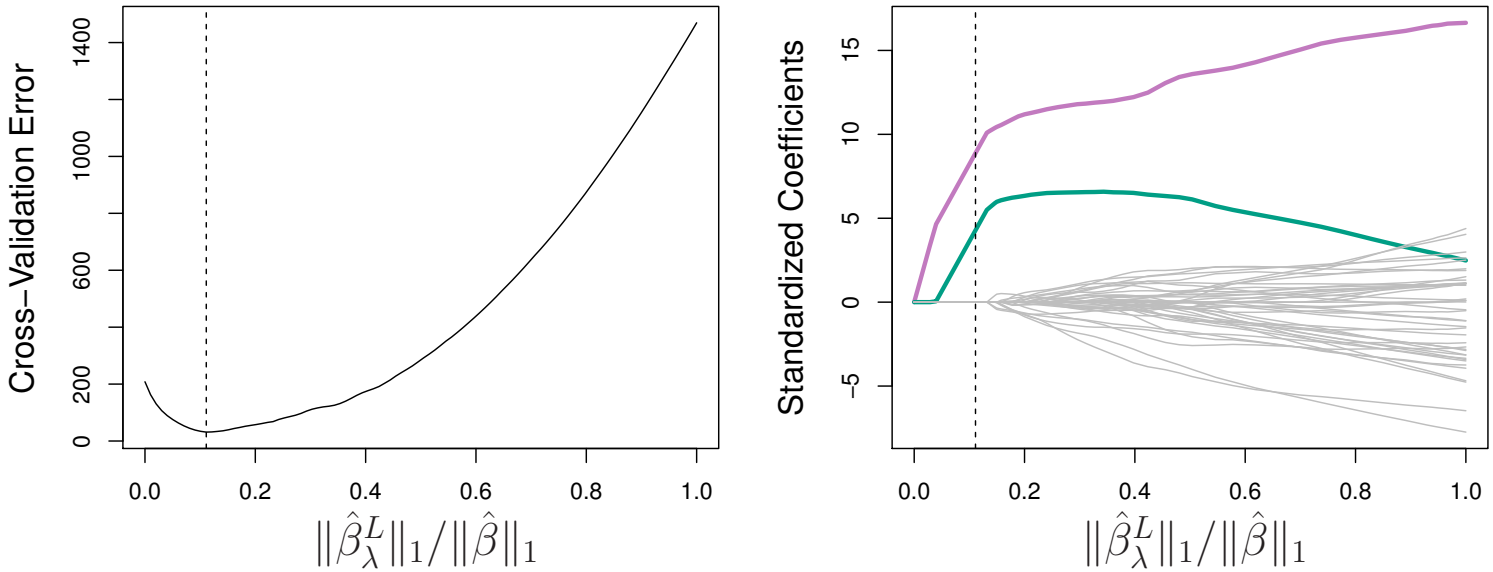


Figure: Lasso Cross-Validation

Left: Ten-fold cross-validation MSE for the lasso on sparse simulated data. **Right:** Corresponding lasso coefficient estimates. The vertical dashed lines indicate the lasso fit with smallest cross-validation error.