

Yale University

Regularized Regression and Model Selection

S&DS 265 · Introductory Machine Learning · Lecture 4

Zhuoran Yang

Fall 2026

AI usage: figures were generated, and these slides polished, with the help of AI tools.

Outline

1. Unstable Linear Fits
2. Ridge Regression
3. The Lasso
4. Cross-Validation and Model Selection

Learning Objectives

In this lecture you will learn:

1. Why similar training fits can produce unstable coefficients and different predictions;
2. How ridge regression trades a small amount of bias for stable, unique estimates;
3. Why the lasso can set coefficients exactly to zero and when that selection is unstable;
4. How cross-validation selects a penalty without using the final test set;
5. How coefficient comparisons and held-out prediction reveal different strengths of ridge and the lasso.

Outline

1. Unstable Linear Fits
2. Ridge Regression
3. The Lasso
4. Cross-Validation and Model Selection

Motivating Example: Credit Card Balances

A bank has records on 400 credit-card customers: income, the credit limit and rating it assigned them, number of cards held, and age of customers.

It also knows the balance each carries.

A new customer applies. What balance should the bank expect, and which of these facts actually matters?

Example and data: ISLP Chapter 6 (James, Witten, Hastie, Tibshirani, and Taylor, 2023).

Credit Data

row	income	limit	rating	cards	age	balance
1	14.891	3606	283	2	34	333
2	106.025	6645	483	3	82	903
3	104.593	7075	514	4	71	580
4	148.924	9504	681	3	36	964

One row is one customer. Income is in thousands of dollars; limit and balance in dollars; rating is a credit score; cards a count; age in years.

Source: The Credit data, ISLP Chapter 6; 400 customers.

Reading One Row

row	income	limit	rating	cards	age	balance
1	14.891	3606	283	2	34	333

This customer earns \$14,891 a year. The bank gave them a \$3,606 credit limit and a credit rating of 283; they hold two cards and are 34 years old. They carry a \$333 balance.

That last number is the one we want to predict for someone new.

Problem Setup

Training data are $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$, with

$$x_i \in \mathbb{R}^p, \quad y_i \in \mathbb{R}.$$

A linear predictor is $f_\beta(x) = \beta_0 + x^\top \beta$. We seek accurate predictions at a new x^* while keeping the fitted rule stable across plausible samples.

For `Credit`, $n = 400$, y_i is balance, and the initial numerical feature set has $p = 6$: income, limit, rating, cards, age, and education.

Correlations Among the Features

Before fitting anything, look at how the features relate to each other.

	income	limit	rating	cards	age	educ.
income	1.00	.79	.79	-.02	.18	-.03
limit	.79	1.00	.997	.01	.10	-.02
rating	.79	.997	1.00	.05	.10	-.03
cards	-.02	.01	.05	1.00	.04	-.05
age	.18	.10	.10	.04	1.00	.00
educ.	-.03	-.02	-.03	-.05	.00	1.00

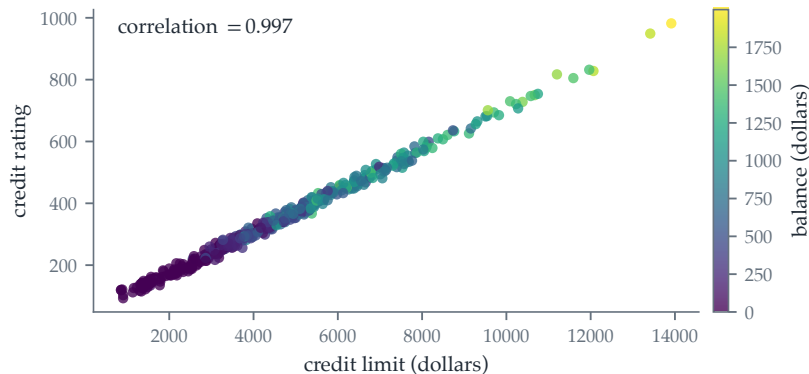
Correlations Among the Features

Before fitting anything, look at how the features relate to each other.

	income	limit	rating	cards	age	educ.
income	1.00	.79	.79	-.02	.18	-.03
limit	.79	1.00	.997	.01	.10	-.02
rating	.79	.997	1.00	.05	.10	-.03
cards	-.02	.01	.05	1.00	.04	-.05
age	.18	.10	.10	.04	1.00	.00
educ.	-.03	-.02	-.03	-.05	.00	1.00

One entry stands out: **limit and rating correlate at .997**. The bank sets a limit largely **from the rating**.

Limit and Rating Say the Same Thing



A customer's credit **rating** is very nearly a rescaling of their credit **limit**: the two carry almost identical information about a customer.

Source: ISLP Credit data; colour records balance.

The Coefficients Change Across Samples

Split the 400 customers in half and refit the same model.

	limit	rating
first half of customers	0.195	1.14
second half of customers	0.063	2.90

The rating coefficient **doubles**; the limit coefficient falls by two thirds. Only the sampled customers changed.

The Coefficients Change Across Samples

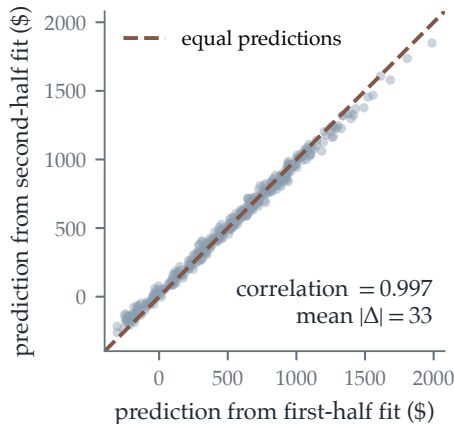
Split the 400 customers in half and refit the same model.

	limit	rating
first half of customers	0.195	1.14
second half of customers	0.063	2.90

The rating coefficient **doubles**; the limit coefficient falls by two thirds. Only the sampled customers changed.

This table establishes coefficient instability. It does not yet tell us whether the two fitted rules make different predictions.

But Their Predictions Remain Close



Both half-sample models predict the [same 400 rows](#). Near the dashed 45° line means nearly equal predictions. Mean absolute difference: \$33; observed-balance SD: \$460.

ISLP Credit data; OLS on rows 1–200 versus 201–400.

Collinearity, Defined

- ▶ **Collinear:** one column is a combination of the others, so some $v \neq 0$ has $Xv = 0$.
Then $\text{rank}(X) < d$.
- ▶ **Nearly collinear:** some $v \neq 0$ makes $\|Xv\|_2$ merely **small**.

Collinearity, Defined

- ▶ **Collinear:** one column is a combination of the others, so some $v \neq 0$ has $Xv = 0$. Then $\text{rank}(X) < d$.
- ▶ **Nearly collinear:** some $v \neq 0$ makes $\|Xv\|_2$ merely **small**.

Why that is a problem: for any scalar c ,

$$X(\hat{\beta} + cv) = X\hat{\beta} + cXv \approx X\hat{\beta},$$

so $\hat{\beta} + cv$ fits the data essentially as well as $\hat{\beta}$. A whole **ridge** of coefficient vectors is nearly optimal, and the data has no basis for preferring one.

For `Credit`, v is roughly “add to limit, subtract from rating”—which is exactly the trade we saw between the two halves.

The Condition Number

Along an eigendirection v of $X^T X$ with eigenvalue λ ,

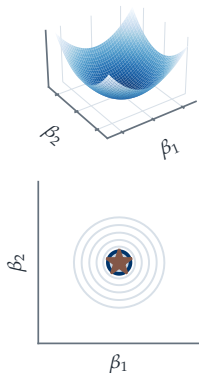
$$v^T X^T X v = \|Xv\|_2^2 = \lambda,$$

so λ is the curvature of the loss in that direction. Near collinearity means some λ is nearly zero: the loss is almost flat there. The **condition number** $\kappa = \lambda_{\max}/\lambda_{\min} \geq 1$ measures how **ill-conditioned** the problem is.

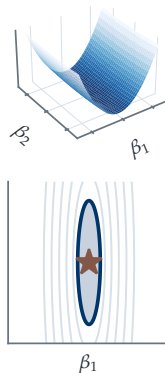
A level set of the loss has semi-axes proportional to $1/\sqrt{\lambda}$, so it is $\sqrt{\kappa}$ times longer than it is wide. For standardized Credit, $\kappa = 1262$ and the valley is about $\sqrt{1262} \approx 36$ times longer than wide.

What Ill-Conditioning Looks Like

$\kappa = 1$: a round bowl



$\kappa = 30$: a narrow valley



Shaded: parameters within ϵ of the minimum. A narrow valley leaves a long sliver the data cannot distinguish within.

Illustration of quadratic losses with the stated condition numbers.

Three Consequences of One Number

On standardized Credit, $\kappa(X^\top X) = 1262$.

Three Consequences of One Number

On standardized Credit, $\kappa(X^\top X) = 1262$.

1. **Statistical.** The variance of $\hat{\beta}$ along the flattest direction scales like $1/\lambda_{\min}$: coefficients become unstable.
2. **Numerical.** Solving the normal equations can amplify a perturbation in $X^\top y$ by as much as κ .
3. **Computational.** The Hessian has the same κ ; steepest descent zigzags across the valley and crawls along it.

Do not form and invert $X^\top X$: use `numpy.linalg.lstsq`. Lecture 5 takes up the computational consequence.

Aside: The Pseudo-Inverse

When $X^T X$ is singular the normal equations still have solutions—just infinitely many. The **Moore–Penrose pseudo-inverse** X^+ picks one of them:

$$\hat{\beta}^+ = X^+ y \quad \text{is the solution of smallest norm } \|\beta\|_2.$$

It is what `numpy.linalg.lstsq` returns in the singular case.

It is worth being clear about what this does and does not do. It resolves the ambiguity by a **convention**— $\hat{\beta}^+$ has the smallest norm. The data still cannot distinguish $\hat{\beta}^+$ from the other minimizers, so the coefficients remain uninterpretable. Regularization makes the same kind of choice, but states it in the objective and lets us tune it.

Aside: Why Features Need a Common Scale

The Credit columns are measured in unrelated units:

	income	limit	rating	cards	age	educ.
standard deviation	35	2308	155	1.4	17	3.1

Aside: Why Features Need a Common Scale

The Credit columns are measured in unrelated units:

	income	limit	rating	cards	age	educ.
standard deviation	35	2308	155	1.4	17	3.1

Two things break as a result.

1. A penalty like $\|\beta\|_2^2$ adds coefficients in different units, so it restrains the large-scale columns far less.
2. The conditioning is wrecked for no real reason: $\kappa(X^\top X) = 2.1 \times 10^7$ on the raw columns.

Aside: Standardizing the Design Matrix

Work **column by column**. For each feature j :

1. Compute its mean μ_j and standard deviation s_j over the rows;
2. Replace every entry by $\tilde{x}_{ij} = (x_{ij} - \mu_j)/s_j$.

Every column of $\tilde{X}$ then has mean 0 and standard deviation 1, so no feature is large merely because of its units. On `Credit` this takes $\kappa(X^\top X)$ from 2.1×10^7 down to 1262.

Aside: Standardizing the Design Matrix

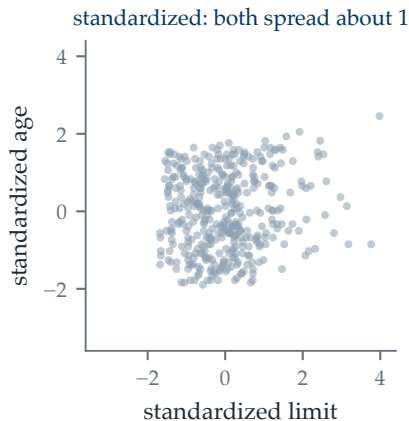
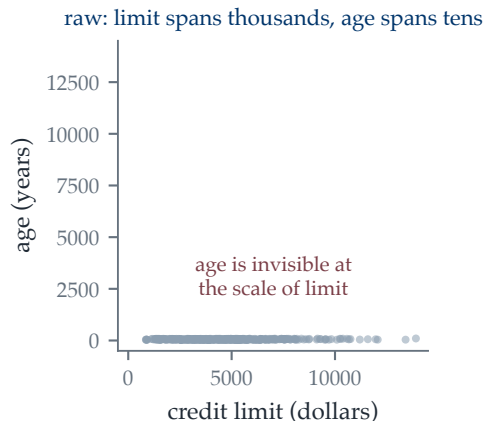
Work **column by column**. For each feature j :

1. Compute its mean μ_j and standard deviation s_j over the rows;
2. Replace every entry by $\tilde{x}_{ij} = (x_{ij} - \mu_j)/s_j$.

Every column of $\tilde{X}$ then has mean 0 and standard deviation 1, so no feature is large merely because of its units. On `Credit` this takes $\kappa(X^\top X)$ from 2.1×10^7 down to 1262.

Compute μ_j, s_j on the training fold only: using full-data statistics leaks held-out information. The intercept is fitted separately and never penalized.

Aside: What Standardizing Does



On common axes, raw age variation disappears beside limit. After standardizing, both features have comparable scale.

ISLP Credit data.

Outline

1. Unstable Linear Fits
2. Ridge Regression
3. The Lasso
4. Cross-Validation and Model Selection

Ridge Objective

Ridge regression minimizes

$$L_{\lambda}(\beta) = \frac{1}{2n} \|y - X\beta\|_2^2 + \frac{\lambda}{2} \|\beta\|_2^2, \quad \lambda \geq 0.$$

The first term rewards fitting the data; the second charges for the size of the coefficient vector.

Ridge Objective

Ridge regression minimizes

$$L_{\lambda}(\beta) = \frac{1}{2n} \|y - X\beta\|_2^2 + \frac{\lambda}{2} \|\beta\|_2^2, \quad \lambda \geq 0.$$

The first term rewards fitting the data; the second charges for the size of the coefficient vector.

- ▶ $\lambda = 0$ recovers ordinary least squares exactly.
- ▶ As λ grows, comparably fitting vectors of **smaller norm** are preferred.
- ▶ As $\lambda \rightarrow \infty$ every coefficient is driven to zero.

The next slides quantify what this buys and costs.

Penalized and Constrained Forms

The penalized problem

$$\min_{\beta} \frac{1}{2n} \|y - X\beta\|_2^2 + \frac{\lambda}{2} \|\beta\|_2^2$$

has a corresponding constrained form

$$\min_{\beta} \frac{1}{2n} \|y - X\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_2^2 \leq t.$$

Penalized and Constrained Forms

The penalized problem

$$\min_{\beta} \frac{1}{2n} \|y - X\beta\|_2^2 + \frac{\lambda}{2} \|\beta\|_2^2$$

has a corresponding constrained form

$$\min_{\beta} \frac{1}{2n} \|y - X\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_2^2 \leq t.$$

These are two views of one problem: for every $\lambda > 0$ there is a $t(\lambda)$ for which the two have the **same** solution, and conversely. Larger λ corresponds to smaller t .

The formal statement is a duality result in convex optimization; we use only the correspondence. The constrained form is the easier one to draw, which is why the geometry pictures use it.

Ridge Normal Equations

Differentiate the two terms:

$$\nabla_{\beta} L_{\lambda}(\beta) = \frac{1}{n} X^{\top} (X\beta - y) + \lambda\beta.$$

Set the gradient to zero and multiply by n :

$$(X^{\top} X + n\lambda I) \hat{\beta}_{\lambda} = X^{\top} y.$$

Differentiate the two terms:

$$\nabla_{\beta} L_{\lambda}(\beta) = \frac{1}{n} X^{\top} (X\beta - y) + \lambda\beta.$$

Set the gradient to zero and multiply by n :

$$(X^{\top}X + n\lambda I)\hat{\beta}_{\lambda} = X^{\top}y.$$

Therefore

$$\hat{\beta}_{\lambda} = (X^{\top}X + n\lambda I)^{-1}X^{\top}y.$$

For every $\lambda > 0$, the matrix is invertible even when $X^{\top}X$ is singular.

Ridge with a Single Feature

Take one standardized feature $x \in \mathbb{R}^n$, so $x^\top x = n$. The objective is

$$L_\lambda(\beta) = \frac{1}{2n} \|y - x\beta\|_2^2 + \frac{\lambda}{2} \beta^2,$$

with derivative

$$-\frac{1}{n} x^\top (y - x\beta) + \lambda\beta = -\frac{x^\top y}{n} + \beta + \lambda\beta.$$

Ridge with a Single Feature

 derive

Take one standardized feature $x \in \mathbb{R}^n$, so $x^\top x = n$. The objective is

$$L_\lambda(\beta) = \frac{1}{2n} \|y - x\beta\|_2^2 + \frac{\lambda}{2} \beta^2,$$

with derivative

$$-\frac{1}{n} x^\top (y - x\beta) + \lambda\beta = -\frac{x^\top y}{n} + \beta + \lambda\beta.$$

Setting this to zero, and writing $\hat{\beta} = x^\top y/n$ for the least-squares coefficient,

$$\hat{\beta}_\lambda = \frac{\hat{\beta}}{1 + \lambda}.$$

Ridge divides the least-squares coefficient by $1 + \lambda$: the same direction, pulled toward zero by a factor we control.

Ridge Reduces the Condition Number

Ridge solves $(X^T X + n\lambda I)\hat{\beta}_\lambda = X^T y$. Adding $n\lambda I$ shifts **every** eigenvalue up by the same amount, so

$$\kappa_\lambda = \frac{\lambda_{\max} + n\lambda}{\lambda_{\min} + n\lambda} \leq \frac{\lambda_{\max}}{\lambda_{\min}} = \kappa,$$

and the improvement is largest exactly where the trouble was: directions where $X^T X$ has small eigenvalues.

Ridge Reduces the Condition Number

Ridge solves $(X^\top X + n\lambda I)\hat{\beta}_\lambda = X^\top y$. Adding $n\lambda I$ shifts **every** eigenvalue up by the same amount, so

$$\kappa_\lambda = \frac{\lambda_{\max} + n\lambda}{\lambda_{\min} + n\lambda} \leq \frac{\lambda_{\max}}{\lambda_{\min}} = \kappa,$$

and the improvement is largest exactly where the trouble was: directions where $X^\top X$ has small eigenvalues.

penalty	none	10	100	1000
κ on standardized Credit	1262	102	11.9	2.1

The objective also becomes strongly convex, so the minimizer is unique and gradient descent converges at a rate set by κ_λ .

Ridge on the Credit Data

Standardized features and centred response; $\text{Cor}(\text{Limit}, \text{Rating}) = .997$.

	least squares	ridge
Income	-266.0	-91.3
Limit	290.1	209.2
Rating	318.8	209.2
Cards	15.9	20.7
Age	-15.4	-21.4
Education	6.2	4.9

Ridge on the Credit Data

Standardized features and centred response; $\text{Cor}(\text{Limit}, \text{Rating}) = .997$.

	least squares	ridge
Income	-266.0	-91.3
Limit	290.1	209.2
Rating	318.8	209.2
Cards	15.9	20.7
Age	-15.4	-21.4
Education	6.2	4.9

Ridge gives the collinear pair **essentially the same coefficient**: it splits their shared signal rather than choosing between nearly identical columns.

Let $X^T X = V \text{diag}(d_1, \dots, d_p) V^T$. In the eigenvector basis,

$$\hat{\beta}_\lambda = V \text{diag}\left(\frac{1}{d_j + n\lambda}\right) V^T X^T y.$$

Relative to least squares, direction v_j is multiplied by

$$\frac{d_j}{d_j + n\lambda}.$$

Small-eigenvalue directions are shrunk most. These are exactly the directions that were least identified by the data.

Recall: Bias, Variance, and MSE

Hold X fixed. Imagine repeatedly drawing new noise, refitting the same procedure, and obtaining a scalar estimate $\tilde{\beta}$ each time.

$$\text{Bias}(\tilde{\beta} \mid X) = \mathbb{E}[\tilde{\beta} \mid X] - \beta^*, \quad \text{Var}(\tilde{\beta} \mid X) = \mathbb{E}[(\tilde{\beta} - \mathbb{E}[\tilde{\beta} \mid X])^2 \mid X].$$

- ▶ **Bias:** systematic displacement of the estimates' centre from truth;
- ▶ **Variance:** sample-to-sample fluctuation around that centre.

Recall: Bias, Variance, and MSE

Hold X fixed. Imagine repeatedly drawing new noise, refitting the same procedure, and obtaining a scalar estimate $\tilde{\beta}$ each time.

$$\text{Bias}(\tilde{\beta} | X) = \mathbb{E}[\tilde{\beta} | X] - \beta^*, \quad \text{Var}(\tilde{\beta} | X) = \mathbb{E}[(\tilde{\beta} - \mathbb{E}[\tilde{\beta} | X])^2 | X].$$

- **Bias:** systematic displacement of the estimates' centre from truth;
- **Variance:** sample-to-sample fluctuation around that centre.

Their combined error is the **mean squared error**:

$$\text{MSE}(\tilde{\beta} | X) := \mathbb{E}[(\tilde{\beta} - \beta^*)^2 | X] = \text{Bias}(\tilde{\beta} | X)^2 + \text{Var}(\tilde{\beta} | X).$$

MSE is the average squared distance from the true coefficient over repeated data sets. Here it measures **estimation error**, not training error.

Ridge Trades Bias for Variance

 derive

With one standardized feature, $x^\top x = n$, and $y = x\beta^* + \varepsilon$, where $\mathbb{E}\varepsilon = 0$ and $\text{Var}(\varepsilon) = \sigma^2 I$. Least squares gives

$$\hat{\beta} = \frac{x^\top y}{n} = \beta^* + \frac{x^\top \varepsilon}{n}, \quad \mathbb{E}\hat{\beta} = \beta^*, \quad \text{Var}(\hat{\beta}) = \frac{\sigma^2}{n}.$$

Ridge Trades Bias for Variance

 derive

With one standardized feature, $x^\top x = n$, and $y = x\beta^* + \varepsilon$, where $\mathbb{E}\varepsilon = 0$ and $\text{Var}(\varepsilon) = \sigma^2 I$. Least squares gives

$$\hat{\beta} = \frac{x^\top y}{n} = \beta^* + \frac{x^\top \varepsilon}{n}, \quad \mathbb{E}\hat{\beta} = \beta^*, \quad \text{Var}(\hat{\beta}) = \frac{\sigma^2}{n}.$$

Ridge shrinks this estimate: $\hat{\beta}_\lambda = \hat{\beta}/(1 + \lambda)$. Therefore

$$\text{Bias}(\hat{\beta}_\lambda | X) = \frac{-\lambda\beta^*}{1 + \lambda}, \quad \text{Var}(\hat{\beta}_\lambda | X) = \frac{\sigma^2/n}{(1 + \lambda)^2}.$$

Ridge Trades Bias for Variance

 derive

With one standardized feature, $x^\top x = n$, and $y = x\beta^* + \varepsilon$, where $\mathbb{E}\varepsilon = 0$ and $\text{Var}(\varepsilon) = \sigma^2 I$. Least squares gives

$$\hat{\beta} = \frac{x^\top y}{n} = \beta^* + \frac{x^\top \varepsilon}{n}, \quad \mathbb{E}\hat{\beta} = \beta^*, \quad \text{Var}(\hat{\beta}) = \frac{\sigma^2}{n}.$$

Ridge shrinks this estimate: $\hat{\beta}_\lambda = \hat{\beta}/(1 + \lambda)$. Therefore

$$\text{Bias}(\hat{\beta}_\lambda | X) = \frac{-\lambda\beta^*}{1 + \lambda}, \quad \text{Var}(\hat{\beta}_\lambda | X) = \frac{\sigma^2/n}{(1 + \lambda)^2}.$$

Increasing λ pulls the centre toward zero (more bias) but makes the estimate move less across samples (less variance).

In This Example, the Best Penalty Is Positive



Substitute the ridge bias and variance into $\text{MSE} = \text{Bias}^2 + \text{Var}$:

$$\text{MSE}(\lambda) = \mathbb{E}[(\hat{\beta}_\lambda - \beta^*)^2 | X] = \frac{\lambda^2(\beta^*)^2 + \sigma^2/n}{(1 + \lambda)^2}.$$

In This Example, the Best Penalty Is Positive

 derive

Substitute the ridge bias and variance into $\text{MSE} = \text{Bias}^2 + \text{Var}$:

$$\text{MSE}(\lambda) = \mathbb{E}[(\hat{\beta}_\lambda - \beta^*)^2 | X] = \frac{\lambda^2(\beta^*)^2 + \sigma^2/n}{(1 + \lambda)^2}.$$

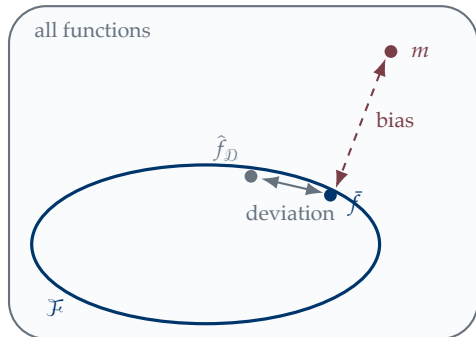
For $\beta^* \neq 0$, differentiating gives

$$\lambda^* = \frac{\sigma^2}{n(\beta^*)^2} > 0.$$

- ▶ More noise or less data calls for **more** shrinkage; a stronger signal calls for **less**.
- ▶ Because β^* and σ^2 are unknown, we cannot use this formula directly. We will choose λ by held-out prediction.

If $\beta^* = 0$, complete shrinkage is optimal; either way, lower variance can outweigh squared bias.

The Same Ideas Apply to Prediction Rules



m the truth we want to predict.

$\bar{f}$ the average rule $\mathbb{E}_D[\hat{f}_D]$. Its displacement from m is the **bias**.

$\hat{f}_D$ one fitted rule. Across samples, the fits scatter around $\bar{f}$; their average squared scatter is the **variance**.

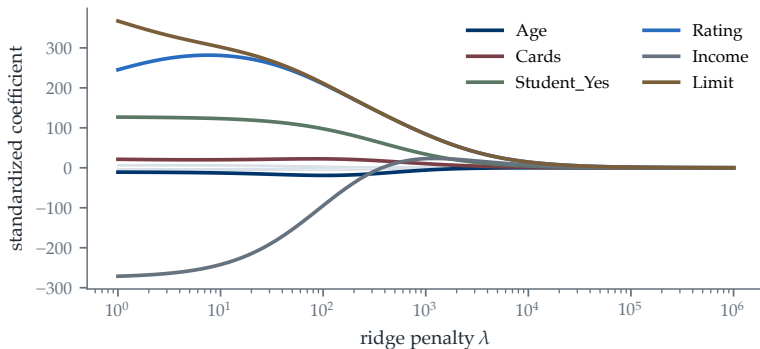
General Pattern of Regularization

The same trade appears everywhere in this course, and it is really a statement about how large a function class we can afford.

- ▶ **Large** $\lambda \Rightarrow$ effectively a **smaller** class of models $\Rightarrow$ larger bias, smaller variance.
- ▶ **Small** $\lambda \Rightarrow$ effectively a **larger** class $\Rightarrow$ smaller bias, larger variance.

With finitely many observations, a richer class fits the truth better but is estimated worse. That is the **bias–variance tradeoff**, and λ is simply a dial on it.

Ridge Coefficient Path



Source: ISLP Credit data; standardized predictors.

Ridge = Weight Decay

Nothing about the penalty $\frac{\lambda}{2}\|\beta\|_2^2$ required the model to be linear. For **any** model f_θ with parameters θ , we can equally well minimize

$$\widehat{R}_n(f_\theta) + \frac{\lambda}{2}\|\theta\|_2^2, \quad \|\theta\|_2^2 = \sum_j \theta_j^2$$

summed over every parameter of the model.

Ridge = Weight Decay

Nothing about the penalty $\frac{\lambda}{2}\|\theta\|_2^2$ required the model to be linear. For **any** model f_θ with parameters θ , we can equally well minimize

$$\widehat{R}_n(f_\theta) + \frac{\lambda}{2}\|\theta\|_2^2, \quad \|\theta\|_2^2 = \sum_j \theta_j^2$$

summed over every parameter of the model.

In deep learning this is called **weight decay**, and it is the standard regularizer for networks with millions of parameters. It is the same idea we have just written down for a linear model: among parameter settings that fit comparably, prefer the one of smaller norm.

Outline

1. Unstable Linear Fits
2. Ridge Regression
3. The Lasso
4. Cross-Validation and Model Selection

Recall: The Price of Each Parameter

Lecture 3 showed that under $y = X\beta^* + \varepsilon$ with $\text{Var}(\varepsilon) = \sigma^2 I$ and X of full column rank d , the in-sample prediction error of least squares is exactly

$$\frac{1}{n} \mathbb{E} \|X(\hat{\beta} - \beta^*)\|_2^2 = \sigma^2 \frac{d}{n}.$$

Every parameter we estimate costs σ^2/n : the error falls as n grows and rises in proportion to the number of features.

What Happens When d Approaches n

Read $\sigma^2 d/n$ at the extremes:

$d = 5, n = 400$	error $\approx 0.013 \sigma^2$: negligible
------------------	---

$d = 200, n = 400$	error $\approx 0.5 \sigma^2$: substantial
--------------------	--

$d \geq n$	the bound is vacuous, and $\hat{\beta}$ is not even unique
------------	--

With $d > n$ there is genuinely no way to estimate an arbitrary vector in $\mathbb{R}^d$ from n observations: the data constrain β only on an n -dimensional subspace and say nothing about the remaining $d - n$ directions.

No estimator escapes this. The only way forward is to **assume that β^* is simpler than a general vector.**

The Sparsity Assumption

The standard assumption is **sparsity**: of the d coefficients only s are nonzero, with $s \ll n$. The signal is **low-dimensional but sits in a high-dimensional space**.

- ▶ If we knew **which** s coordinates were nonzero, the problem would collapse to a fit in $\mathbb{R}^s$, costing $\sigma^2 s/n$.
- ▶ We do not know them, so the support must be found from the data.
- ▶ The lasso finds it and estimates on it at once, at a cost of roughly $\sigma^2 s \log(d)/n$ — only a $\log d$ penalty for not knowing.

Sparsity is an assumption about the world, like linearity. If every coefficient is small but nonzero, ridge is the better tool.

Lasso Objective

The lasso replaces the squared penalty by an absolute-value penalty:

$$L_{\lambda}(\beta) = \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1, \quad \|\beta\|_1 = \sum_{j=1}^p |\beta_j|.$$

Lasso Objective

The lasso replaces the squared penalty by an absolute-value penalty:

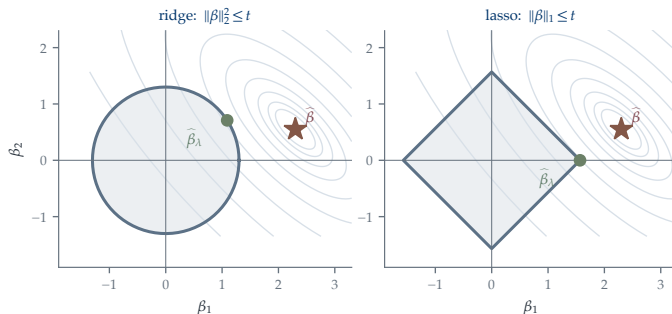
$$L_{\lambda}(\beta) = \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1, \quad \|\beta\|_1 = \sum_{j=1}^p |\beta_j|.$$

The objective is convex but not differentiable when any $\beta_j = 0$.

The kink is the mechanism that makes a coefficient exactly zero; it is not a numerical accident.

It also means there is no closed form, and no gradient at zero, so the lasso needs a genuinely different algorithm. We will introduce it in Lecture 5.

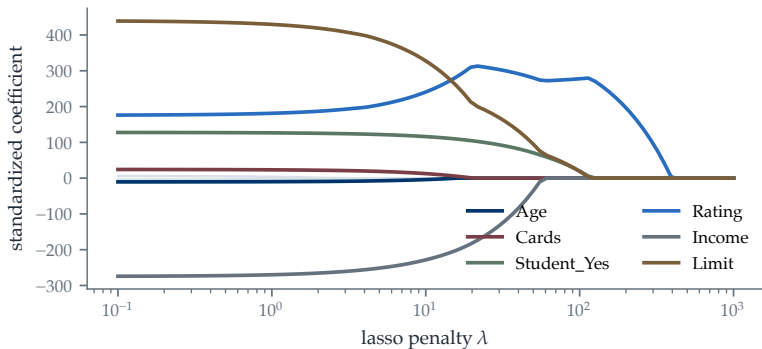
Why the Lasso Produces Exact Zeros



The solution is where an $\hat{R}_n$ contour first touches the constraint. The ridge ball is smooth; the lasso diamond has **axis corners**. Contact at one makes $\hat{\beta}_2$ exactly zero.

Source: Original two-parameter quadratic example.

Lasso Coefficient Path



Source: ISLP Credit data; standardized predictors.

Lasso on the Credit Data

The same standardized Credit features, now with an ℓ_1 penalty.

	least squares	ridge	lasso
Income	-266.0	-91.3	-230.1
Limit	290.1	209.2	220.2
Rating	318.8	209.2	351.8
Cards	15.9	20.7	6.9
Age	-15.4	-21.4	-9.6
Education	6.2	4.9	0.0

Lasso on the Credit Data

The same standardized Credit features, now with an ℓ_1 penalty.

	least squares	ridge	lasso
Income	-266.0	-91.3	-230.1
Limit	290.1	209.2	220.2
Rating	318.8	209.2	351.8
Cards	15.9	20.7	6.9
Age	-15.4	-21.4	-9.6
Education	6.2	4.9	0.0

Ridge shrinks every coefficient; the lasso **removes** Education, leaving five features rather than six.

Correlated Predictors

When two features carry nearly the same information, ridge and the lasso behave differently:

- ▶ **Ridge** shares the weight across them, as we saw with `Limit` and `Rating` landing at 209.2 apiece;
- ▶ **The lasso** tends to keep one and zero the other, and which one it keeps can change with a small change in the sample.

Correlated Predictors

When two features carry nearly the same information, ridge and the lasso behave differently:

- ▶ **Ridge** shares the weight across them, as we saw with `Limit` and `Rating` landing at 209.2 apiece;
- ▶ **The lasso** tends to keep one and zero the other, and which one it keeps can change with a small change in the sample.

So a nonzero lasso coefficient is a prediction-oriented choice under one sample and one λ . It is not evidence that a feature matters and its correlated twin does not.

Outline

1. Unstable Linear Fits
2. Ridge Regression
3. The Lasso
4. Cross-Validation and Model Selection

A Second Data Set: Baseball Salaries

at-bats	hits	home runs	years	career hits	salary
315	81	7	14	835	475.0
479	130	18	3	457	480.0
496	141	20	11	1575	500.0
⋮	⋮	⋮	⋮	⋮	⋮

One row is one player, described by 16 counting statistics: at-bats (turns batting), hits, home runs, walks, years, and career totals. The response is [log salary](#) in thousands. Row 1 batted 315 times for 81 hits, had played 14 years for 835 career hits, and was paid [\\$475,000](#).

Source: The Hitters data, ISLP Chapter 6; 263 players with complete records.

Which λ Should We Choose?

Ridge and the lasso both leave one number undetermined — regularization parameter λ .

We showed that some $\lambda > 0$ beats least squares, but the best value depended on σ^2 and β^* , which we do not know.

Which λ Should We Choose?

Ridge and the lasso both leave one number undetermined — regularization parameter λ .

We showed that some $\lambda > 0$ beats least squares, but the best value depended on σ^2 and β^* , which we do not know.

The obvious idea is to pick the λ that fits the training data best. It fails immediately:

$$\widehat{R}_n(\hat{\beta}_{\lambda=0}) \leq \widehat{R}_n(\hat{\beta}_\lambda) \quad \text{for every } \lambda > 0,$$

because least squares already minimizes $\widehat{R}_n$ over **all** coefficient vectors. Training risk always votes for $\lambda = 0$.

Judge the Model the Way It Will Be Used

Think about what the fitted rule is **for**. It will be applied to players whose salaries are not yet set, and to applicants the bank has never seen. Performance on those future cases is the only thing that matters.

Judge the Model the Way It Will Be Used

Think about what the fitted rule is **for**. It will be applied to players whose salaries are not yet set, and to applicants the bank has never seen. Performance on those future cases is the only thing that matters.

So λ should be chosen by how well the model predicts data it did not fit. We have no future data—but we can **manufacture** the situation: hold part of our data back, fit without it, and treat it as though it were unseen.

Judge the Model the Way It Will Be Used

Think about what the fitted rule is **for**. It will be applied to players whose salaries are not yet set, and to applicants the bank has never seen. Performance on those future cases is the only thing that matters.

So λ should be chosen by how well the model predicts data it did not fit. We have no future data—but we can **manufacture** the situation: hold part of our data back, fit without it, and treat it as though it were unseen.

Doing this once wastes data and depends on which rows we happened to hold out. Doing it repeatedly, so every row is held out exactly once, is **cross-validation.**

Five-Fold Cross-Validation

fold 1	validate	fit	fit	fit	fit
fold 2	fit	validate	fit	fit	fit
fold 3	fit	fit	validate	fit	fit
fold 4	fit	fit	fit	validate	fit
fold 5	fit	fit	fit	fit	validate

each row is one split: fit on the four blue blocks, score on the red one

The Cross-Validation Algorithm

Fix a grid of candidate penalties. For **each** candidate λ :

1. For $k = 1, \dots, 5$: fit the model on the other four folds, and score its predictions on fold k . Call that score $s_k(\lambda)$.
2. Average them into $\widehat{R}_{\text{CV}}(\lambda) = \frac{1}{5} \sum_{k=1}^5 s_k(\lambda)$.

The Cross-Validation Algorithm

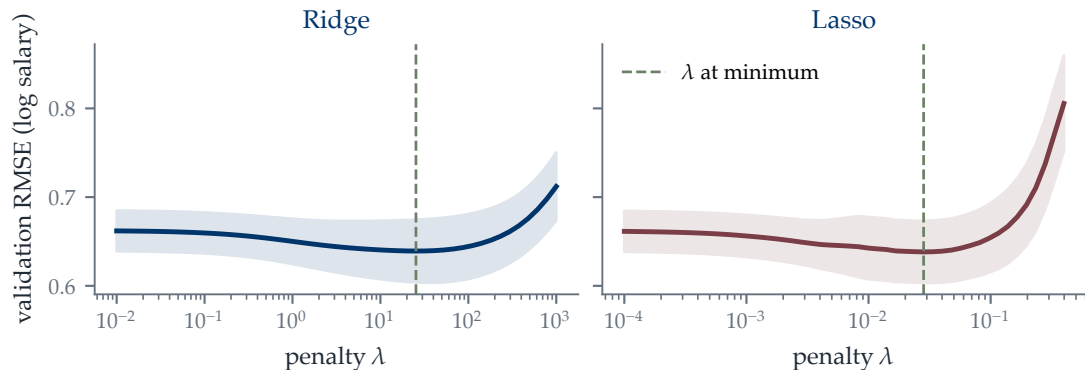
Fix a grid of candidate penalties. For **each** candidate λ :

1. For $k = 1, \dots, 5$: fit the model on the other four folds, and score its predictions on fold k . Call that score $s_k(\lambda)$.
2. Average them into $\hat{R}_{CV}(\lambda) = \frac{1}{5} \sum_{k=1}^5 s_k(\lambda)$.

Then select the λ for which $\hat{R}_{CV}$ is **smallest**, and refit the model at that λ on all the training data.

Every fit uses 80% of the training data and is scored on the 20% it never saw, so $\hat{R}_{CV}(\lambda)$ estimates how that λ would do on new data.

Cross-Validation Curves



The dip is not sharp: a **wide range** of λ gives very similar error. The band shows the spread of the five fold scores, so the curve is flat relative to how precisely we can measure it.

ISLP Hitters training split; five folds, seed 26505.

Choosing λ

The chosen penalty is read straight off the curve: take the λ at which $\widehat{R}_{CV}$ is smallest.

	ridge	lasso
selected λ	25.4	0.028
validation RMSE there	0.639	0.638

At $\lambda = 0.028$ the lasso keeps 7 of the 16 features, so the penalty has done some selection on its own.

Source: Selecting the tuning parameter, ISL §6.2.3.

Final Test Protocol

Once the model family and the penalty $\hat{\lambda}$ are settled:

1. Refit the model at $\hat{\lambda}$ on **all** the training data;
2. Make one prediction for each untouched test observation;
3. Report the test error.

Here 25% of the 263 `Hitters` rows form the final test set; the remaining 75% supply the five-fold selection loop.

Final Test Protocol

Once the model family and the penalty $\hat{\lambda}$ are settled:

1. Refit the model at $\hat{\lambda}$ on **all** the training data;
2. Make one prediction for each untouched test observation;
3. Report the test error.

Here 25% of the 263 Hitters rows form the final test set; the remaining 75% supply the five-fold selection loop.

The test set stays outside the whole loop. Looking at test performance while choosing λ turns the test set into just another validation set.

Summary: Regularized Objectives

$$\hat{\beta}_{\text{ridge}} = \arg \min_{\beta} \left\{ \frac{1}{2n} \|y - X\beta\|_2^2 + \frac{\lambda}{2} \|\beta\|_2^2 \right\},$$

$$\hat{\beta}_{\text{lasso}} = \arg \min_{\beta} \left\{ \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\}.$$

Ridge shrinks unstable directions continuously. The lasso's non-smooth corner can produce exact zeros through soft thresholding.

Summary: Selection and Evaluation

Cross-validation estimates unseen prediction error from the training data:

$$\widehat{R}_{\text{CV}}(\lambda) = \frac{1}{K} \sum_{k=1}^K \frac{1}{|V_k|} \sum_{i \in V_k} \ell(y_i, \tilde{f}_{\lambda}^{(-k)}(x_i)).$$

We select the λ with the smallest $\widehat{R}_{\text{CV}}$, refit at that λ on all the training data, and use the test set once afterwards.

Next Lecture

Least squares and ridge have closed forms; the lasso does not, and its objective is not differentiable at zero. We now need algorithms that move through parameter space rather than solve one matrix equation.

Lecture 5 — Iterative Methods for Linear Regression derives gradient descent, explains how conditioning controls its speed, introduces minibatch gradients, and supplies the extra step needed by the lasso.

Reading for this lecture: ISLP Chapters 5 and 6.