STAT 24620=FINM 34700, STAT 32950
Multivariate Data Analysis
Lecture 10: Regularization for High-Dimensional Supervised Learning

Jingshu Wang

The University of Chicago

Outline

High dimension as the next challenge

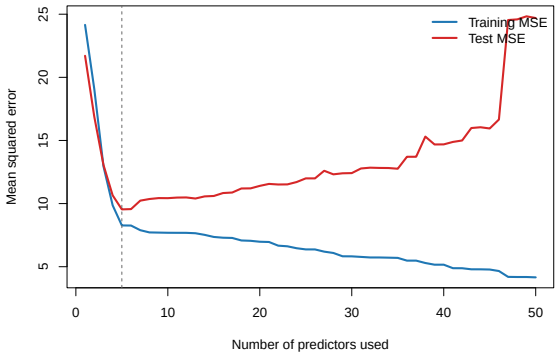
- So far, most methods in the course relied on estimating covariance structure, regression coefficients, or class boundaries in a moderate-dimensional setting.
- Once the number of variables p becomes comparable to n, or even larger than n, these estimators become much less stable.
- This is a general issue, not just a regression issue.

Roadmap:

- today: high-dimensional supervised learning, starting with linear regression;
- next lecture: high-dimensional unsupervised learning, such as sparse PCA and covariance regularization.

Main theme: control model complexity to improve out-of-sample prediction.

A first example: overfitting in linear regression



- Simulated linear-regression example with training size $n = 80$ and test size $n = 1000$.
- We add predictors sequentially: the first few carry signal, later ones are pure noise.
- Even in ordinary linear regression, training error usually decreases as we add more predictors.
- Test error need not decrease: beyond a point, extra flexibility mainly fits noise.

We focus on linear regression first because the central high-dimensional issues already appear here very clearly.

High-dimensional linear regression: what changes?

We begin with the familiar model

$$y = \beta_0 + \beta_1 X_1 + \cdots + \beta_{p-1} X_{p-1} + \varepsilon$$

but now allow the predictor dimension p to be large.

For derivations: after centering, $y = X\beta + \varepsilon$, so β contains only slope coefficients.

What changes?

- Overfitting becomes easier when many predictors are available.
- Estimation becomes unstable when predictors are correlated or when p is not small relative to n.

Symptoms

- large, unstable coefficients and poor test performance;
- $X^\top X$ becomes singular when $p > n$.

What goes wrong with OLS?

Ordinary least squares solves

$$\min_{\beta} \ \|y - X\beta\|_2^2.$$

If $X^\top X$ is invertible,

$$\hat{\beta}_{\text{OLS}} = (X^\top X)^{-1} X^\top y.$$

- If $p > n$, then $X^\top X$ is singular and OLS is not uniquely defined.
- When p is close to n, $X^\top X$ can have small eigenvalues, making the inverse unstable.
- Under the classical model,

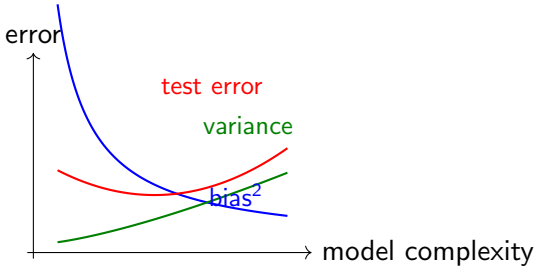
  $$\text{Var}(\hat{\beta}_{\text{OLS}}) = \sigma^2 (X^\top X)^{-1},$$

  so small eigenvalues lead to large variance in $\hat{\beta}_{\text{OLS}}$.

Good in-sample fit can come with unstable coefficients and poor out-of-sample prediction.

Bias-variance intuition

prediction error $\approx$ bias^2 + variance + noise.



- Adding flexibility usually lowers bias.
- But it often increases variance.
- Regularization accepts some bias in exchange for lower variance.

Regularization intentionally trades a little bias for stability and prediction accuracy.

Shrinkage methods at a glance

Ridge

- ℓ_2 penalty;
- shrinks coefficients toward zero;
- keeps all predictors.

Lasso

- ℓ_1 penalty;
- can shrink coefficients exactly to zero;
- performs variable selection.

Elastic net

- combines ℓ_1 and ℓ_2 penalties;
- balances sparsity and stability;
- useful with correlated predictors.

We start with ridge because it is the cleanest way to see how shrinkage stabilizes high-dimensional linear regression.

Ridge regression

Ridge regression solves

$$\min_{\beta} \ \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2, \qquad \lambda \geq 0.$$

Equivalent constrained form

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

- Here β denotes the slope coefficients after centering; the intercept is not penalized.
- λ controls the amount of shrinkage;
- larger λ pushes coefficients toward zero;
- coefficients usually become smaller, but rarely exactly zero.

Deriving the ridge solution

Start from the penalized objective

$$L(\beta) = \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2.$$

Differentiate with respect to β:

$$\frac{\partial L}{\partial \beta} = -2X^\top(y - X\beta) + 2\lambda\beta.$$

Setting the derivative equal to zero gives the ridge normal equations

$$(X^\top X + \lambda I)\beta = X^\top y.$$

So the ridge estimator is

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

Why does ridge help?

- Ridge adds λI to $X^\top X$, pushing all eigenvalues away from zero.
- This stabilizes inversion when predictors are highly correlated or when p is large.
- Ridge is well-defined even when $p > n$.

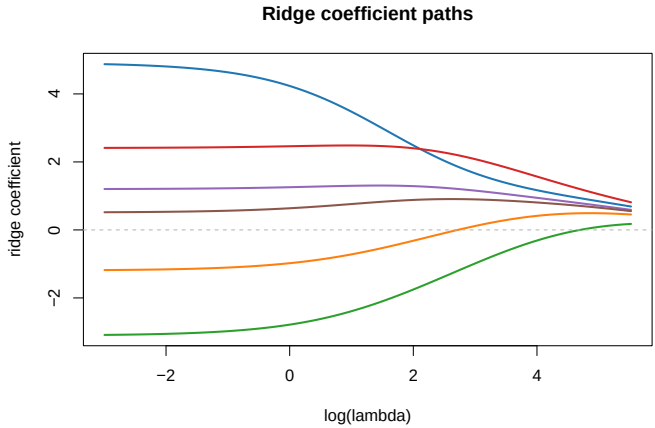
Bias-variance tradeoff

$$\mathbb{E}[\hat{\beta}_{\text{ridge}} \mid X] = (X^\top X + \lambda I)^{-1} X^\top X \, \beta,$$

$$\text{Var}(\hat{\beta}_{\text{ridge}} \mid X) = \sigma^2 (X^\top X + \lambda I)^{-1} X^\top X (X^\top X + \lambda I)^{-1}.$$

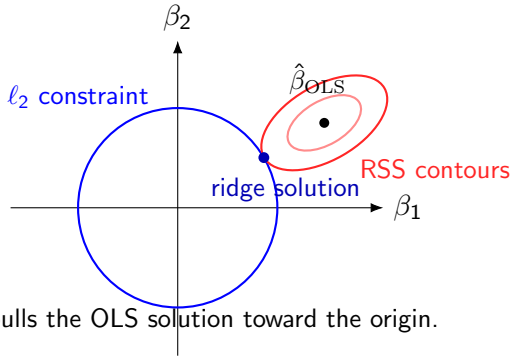
- Larger λ increases bias.
- Larger λ lowers variance.
- The tuning problem is to find the right balance for prediction.

How do $\hat{\beta}_{\text{Ridge}}$ change with λ in practice?



- At $\lambda \approx 0$, ridge is close to OLS.
- As λ increases, coefficients shrink smoothly toward zero.

A geometric view of ridge



Ridge pulls the OLS solution toward the origin.

- OLS chooses the center of the RSS contours, but that point can have a large coefficient norm.
- Ridge restricts us to a ball around the origin, so the solution is a shrunk version of OLS.
- Increasing λ makes the ball smaller and produces more shrinkage toward zero.

Why go beyond ridge?

- Ridge stabilizes estimation, but typically keeps *all* predictors in the model.
- In many applications, only a small subset of variables carries most of the signal.
- Including many variables can make models harder to interpret and communicate.
- We would like a method that can identify and keep only the important predictors.

This leads to the idea of *sparse* models: many coefficients exactly zero.

Lasso

Lasso solves

$$\min_{\beta} \ \|y - X\beta\|_2^2 + \lambda\|\beta\|_1, \qquad \|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|.$$

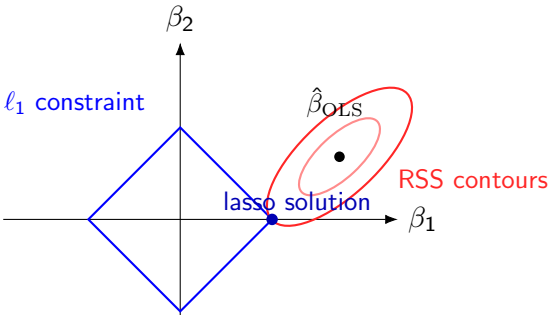
Equivalent constrained form

$$\min_{\beta} \ \|y - X\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_1 \leq t.$$

- As in ridge regression, β denotes only the slope coefficients.
- The ℓ_1 penalty encourages sparsity in the coefficients.
- Like ridge, lasso shrinks coefficients toward zero.
- Unlike ridge, some coefficients are set exactly to zero.

Lasso produces sparse models and performs variable selection automatically.

A geometric view of lasso



Lasso solutions often lie on corners, producing exact zeros.

- The ℓ_1 constraint has corners aligned with the coordinate axes.
- The optimum often occurs at a corner.
- At a corner, at least one coefficient is exactly zero.

Corners in the constraint region are what create sparsity.

A special case: orthogonal design

Suppose

$$X^\top X = I.$$

Then the lasso criterion is

$$\|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$

Expanding the squared loss,

$$\|y - X\beta\|_2^2 = y^\top y - 2\beta^\top X^\top y + \beta^\top X^\top X \beta.$$

Since $X^\top X = I$,

$$\|y - X\beta\|_2^2 = y^\top y - 2\sum_{j=1}^{p} \beta_j \hat{\beta}_j^{\mathrm{OLS}} + \sum_{j=1}^{p} \beta_j^2,$$

where $\hat{\beta}^{\mathrm{OLS}} = X^\top y$.

So the objective separates coordinate by coordinate.

Soft-thresholding in the orthogonal case

For each coordinate, we solve

$$\min_{\beta_j} \ \beta_j^2 - 2\beta_j \hat{\beta}_j^{\mathrm{OLS}} + \lambda|\beta_j|.$$

The solution is

$$\hat{\beta}_j^{\mathrm{lasso}} = S(\hat{\beta}_j^{\mathrm{OLS}}, \lambda/2),$$

where

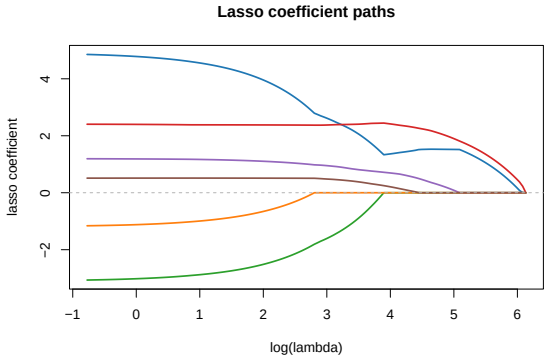
$$S(z,t) = \mathrm{sign}(z)(|z| - t)_+$$

is the soft-thresholding rule.

- If $|\hat{\beta}_j^{\mathrm{OLS}}|$ is small, it is set exactly to zero.
- If $|\hat{\beta}_j^{\mathrm{OLS}}|$ is large, it is shrunk toward zero.
- This explains why lasso produces sparse solutions.

In general, lasso has no closed-form solution, but this special case reveals its basic mechanism.

How do $\hat{\beta}_{\text{Lasso}}$ change with λ in practice?



- As λ increases, coefficients are pulled toward zero.
- Unlike ridge, many lasso paths hit zero at a finite value of λ.
- Larger λ means a sparser model.

Ridge versus lasso

Ridge

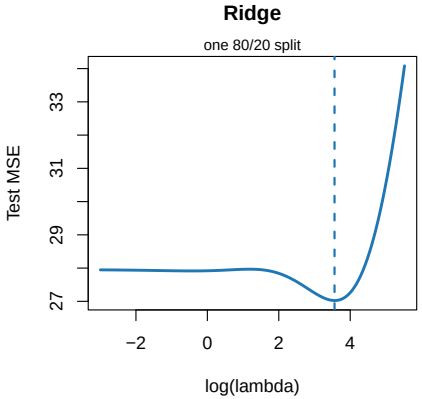
- shrinks all coefficients smoothly;
- usually keeps all predictors;
- often good when many variables have small effects.
- adds bias to reduce variance.

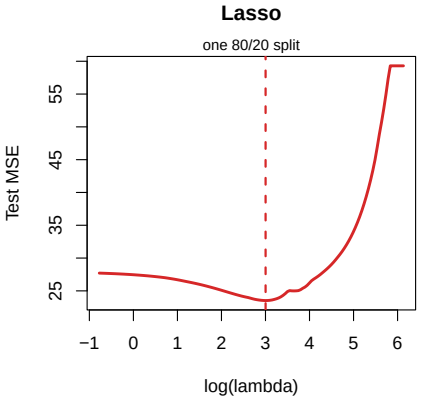
Lasso

- shrinks and selects;
- often yields a sparse model;
- often good when only a few variables matter strongly.
- adds more bias, but may reduce variance much more by removing noise variables.

Both methods are bias-variance tradeoffs: they accept bias in order to reduce instability and improve prediction.

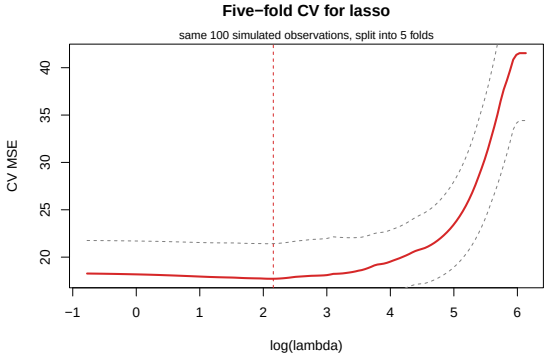
How to choose the tuning parameter λ





- $\lambda = 0$ gives the unregularized fit; large λ overshrinks.
- As λ increases, bias increases but variance decreases.
- Test error is typically U-shaped.
- Choose λ to minimize out-of-sample error.

Choosing λ via cross-validation



- A single train/test split can give a noisy estimate of test error.
- Cross-validation averages over multiple splits of the data.
- For each λ, we average validation error across folds.
- Choose the λ with the lowest estimated error.

What cross-validation computes

Split the data into K folds.

For each λ and each fold k:

$$\hat{f}_\lambda^{(-k)} = \text{prediction model trained without fold } k,$$

$$\mathrm{Err}_k(\lambda) = \frac{1}{|V_k|} \sum_{i \in V_k} \left(y_i - \hat{f}_\lambda^{(-k)}(x_i)\right)^2.$$

Average across folds:

$$\mathrm{CV}(\lambda) = \frac{1}{K} \sum_{k=1}^{K} \mathrm{Err}_k(\lambda).$$

Choose

$$\hat{\lambda} = \arg\min_\lambda \mathrm{CV}(\lambda).$$

What if predictors are highly correlated?

- Ridge distributes weight across correlated predictors.
- Lasso tends to select one predictor and ignore the others.
- When predictors carry similar signal, the selected set can be unstable.

Implication

- variable selection may be unreliable under strong correlation;
- prediction can still be accurate, but interpretation becomes harder.

Can we combine sparsity with stability?

Elastic net: combining ridge and lasso

Elastic net solves

$$\min_{\beta} \ \|y - X\beta\|_2^2 + \lambda\Big(\alpha\|\beta\|_1 + (1-\alpha)\|\beta\|_2^2\Big),$$

where $\alpha \in [0, 1]$.

- $\alpha = 1$ gives the lasso; $\alpha = 0$ gives ridge regression.
- Intermediate values combine sparsity and stability.
- Can select groups of correlated predictors together.

Tuning

- Use cross-validation to choose λ.
- Try a small grid of α values (e.g., 0.1, 0.5, 0.9).

Elastic net trades off sparsity (lasso) and stability (ridge).

Regularized logistic regression

For binary classification, we can also regularize logistic regression.

Let $\ell(\beta_0, \beta)$ denote the log-likelihood. In practice we penalize only the slope vector β:

$$\min_{\beta_0,\beta} \; -\ell(\beta_0, \beta) + \lambda\|\beta\|_1 \qquad \text{or} \qquad -\ell(\beta_0, \beta) + \lambda\|\beta\|_2^2.$$

- same idea as in linear regression: loss + penalty;
- helps when features are numerous or correlated;
- stabilizes estimation and prevents overfitting.

Replace squared loss by logistic loss, keep the same regularization idea.

Practical considerations for regularization

- Penalties act on coefficient magnitudes, so scaling matters.
- Without standardization, variables on larger scales are penalized less.
- Standardize predictors before ridge, lasso, or elastic net.

Typical workflow

- center predictors, and center the response for linear regression;
- scale predictors to comparable variance;
- fit the regularization path;
- use cross-validation to choose λ;
- evaluate out-of-sample performance.

In our derivations, centering lets us omit the intercept; in software, the intercept is typically estimated but not penalized.

An R demo of regularized prediction

Notebook file: `Lecture10_demo.html`

Lecture 10 summary

- High-dimensional prediction requires controlling variance and overfitting.
- Ridge shrinks all coefficients and is stable under collinearity.
- Lasso adds sparsity and variable selection.
- Elastic net combines both ideas.
- The same regularization principle extends naturally to classification models.

Suggested reading

- James, Witten, Hastie, Tibshirani (2nd edition), Chapter 6.2, 6.4, 6.5.2.