Introduction and Regularization

Econometría Aplicada y Ciencia de Datos - CIDE Fall 2026

Carlos Brito

Inspired by material from Francisco Cabrera:

`github.com/fcabrerahz/EconometricsDS`

Review: Basic Matrix Operations

Transpose: swap rows and columns ($3 \times 2 \to 2 \times 3$)

$$X = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \Rightarrow X' = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

Product: $(m \times n)(n \times q) = (m \times q)$, entry = row $\cdot$ column

$$X'X = \begin{pmatrix} 35 & 44 \\ 44 & 56 \end{pmatrix}$$

$(X'X)_{\mathbf{11}} = 1 \cdot 1 + 3 \cdot 3 + 5 \cdot 5 = 35$. $X'X$ is always $k \times k$ and symmetric.

Rank: number of linearly independent columns. $\text{rank}(X) \leq \min(n, k)$.

Determinant and inverse (2×2)

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad \det A = ad - bc$$

$$A^{-\mathbf{1}} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

$AA^{-\mathbf{1}} = I$. Exists only if $\det A \neq 0$; if $\det A \approx 0$.

Eigenvalues: solve $\det(A - \lambda I) = 0$

$$\lambda^{\mathbf{2}} - \text{tr}(A)\,\lambda + \det A = 0$$

- $\sum_j \lambda_j = \text{tr}(A)$, $\prod_j \lambda_j = \det A$.
- Some $\lambda_j = 0 \Leftrightarrow \det A = 0 \Leftrightarrow$ not invertible.
- Eigenvalues of $A^{-\mathbf{1}}$ are $1/\lambda_j$.
- Intuition: $Av = \lambda v \Rightarrow (A - \lambda I)v = 0$ has a nonzero solution only if $A - \lambda I$ is singular.

Before We Start: OLS and Linear Algebra

OLS in matrix form

- $Y = X\beta + \varepsilon$, with X an $n \times p$ matrix $\Rightarrow$ $\hat{\beta}_{OLS} = \arg\min_\beta \|Y - X\beta\|_2^2 = (X'X)^{-1}X'Y$
  - Norms: $\|\beta\|_2 = \left(\sum_k \beta_k^2\right)^{1/2}$, $\|\beta\|_1 = \sum_k |\beta_k|$
- Derive it: first-order condition $X'(Y - X\hat{\beta}) = 0$ (normal equations).
- Gauss–Markov: under the classical assumptions, OLS is unbiased and **BLUE**; $\mathrm{Var}(\hat{\beta} \mid X) = \sigma^2(X'X)^{-1}$.
- Large-sample properties: consistency, asymptotic normality, efficiency.

Linear algebra

- **Rank** and invertibility: $X'X$ is invertible only if $\mathrm{rank}(X) = p \Rightarrow$ fails when $p > n$.
- **Eigenvalues** of $X'X$: small eigenvalues $\Leftrightarrow$ near-collinearity $\Leftrightarrow$ large entries in $(X'X)^{-1} \Rightarrow$ high variance of $\hat{\beta}$.
- **Standardizing** regressors (mean 0, s.d. 1) so coefficients are on a comparable scale.

Outline

- Introduction
- Ridge Regression
- Lasso Estimator
- Elastic Net

Motivation

Locate the intersection between **Applied Econometrics** and **Machine Learning** to solve empirical problems with modern data.

Two cultures:

- **Data modeling** (assumes a stochastic data-generating process)

Motivation

Locate the intersection between **Applied Econometrics** and **Machine Learning** to solve empirical problems with modern data.

Two cultures:

- **Data modeling** (assumes a stochastic data-generating process)
- **Algorithmic modeling** (treats the data mechanism as unknown)

Motivation

Locate the intersection between **Applied Econometrics** and **Machine Learning** to solve empirical problems with modern data.

Two cultures:

- **Data modeling** (assumes a stochastic data-generating process)
- **Algorithmic modeling** (treats the data mechanism as unknown)

Economics has been committed to the almost exclusive use of data models.

Statistics and ML are now converging; in economics, adoption has been slower.

Motivation

Adoption of ML in economics is important because data have changed:

Motivation

Adoption of ML in economics is important because data have changed:

- **Big data:** we now observe information on a large number of units.
  e.g. IMSS records for millions of formal workers; every card transaction in a city.

Motivation

Adoption of ML in economics is important because data have changed:

- **Big data:** we now observe information on a large number of units.
  e.g. IMSS records for millions of formal workers; every card transaction in a city.
- **Many features**/X for each unit.
  e.g. a job posting described by thousands of words; a firm's full history of invoices.

Motivation

Adoption of ML in economics is important because data have changed:

- **Big data:** we now observe information on a large number of units.
  e.g. IMSS records for millions of formal workers; every card transaction in a city.
- **Many features**/X for each unit.
  e.g. a job posting described by thousands of words; a firm's full history of invoices.
- Often **beyond the simple single–cross-section** setting.
  e.g. panels, networks (who trades with whom), text, and satellite images (night lights to measure GDP).

Motivation

Adoption of ML in economics is important because data have changed:

- **Big data:** we now observe information on a large number of units.
  e.g. IMSS records for millions of formal workers; every card transaction in a city.
- **Many features**/X for each unit.
  e.g. a job posting described by thousands of words; a firm's full history of invoices.
- Often **beyond the simple single–cross-section** setting.
  e.g. panels, networks (who trades with whom), text, and satellite images (night lights to measure GDP).

Move away from dependence on parametric models to a more diverse set of tools, yet:

- Preserve the strengths of applied econometrics (identification, external validity)
- Leverage ML for: selection/regularization, prediction, heterogeneity, and big data

Econometrics vs. Machine Learning

Econometrics:

- Emphasizes **large-sample properties**:

Econometrics vs. Machine Learning

Econometrics:

- Emphasizes **large-sample properties**:
  - Consistency, asymptotic normality, efficiency. Theoretical proofs.

Econometrics vs. Machine Learning

Econometrics:

- Emphasizes **large-sample properties**:
  - Consistency, asymptotic normality, efficiency. Theoretical proofs.

Machine Learning:

- Emphasizes **algorithmic performance**:
  - Practical behavior in specific settings, with an "error-rate" orientation and no asymptotic proofs.

Econometrics vs. Machine Learning

Econometrics:

- Emphasizes **large-sample properties**:
  - Consistency, asymptotic normality, efficiency. Theoretical proofs.

Machine Learning:

- Emphasizes **algorithmic performance**:
  - Practical behavior in specific settings, with an "error-rate" orientation and no asymptotic proofs.
- **No formal proofs**: Are neural networks uniformly superior to methods like regression trees or random forests?

Econometrics vs. Machine Learning

Econometrics:

- Emphasizes **large-sample properties**:
  - Consistency, asymptotic normality, efficiency. Theoretical proofs.

Machine Learning:

- Emphasizes **algorithmic performance**:
  - Practical behavior in specific settings, with an "error-rate" orientation and no asymptotic proofs.
- **No formal proofs**: Are neural networks uniformly superior to methods like regression trees or random forests?

While **valid confidence intervals** are crucial (e.g., estimating an ATE), methods with no formal inference should not be dismissed.

Out-of-sample predictive performance from ML can be valuable, even if prediction is rare in modern econometrics.

Terminology

ML uses new terms for old ideas:

- Estimation: Training

Terminology

ML uses new terms for old ideas:

- Estimation: Training
- Regressors: Features

Terminology

ML uses new terms for old ideas:

- Estimation: Training
- Regressors: Features
- Parameters: Weights

Terminology

ML uses new terms for old ideas:

- Estimation: Training
- Regressors: Features
- Parameters: Weights
- Supervised learning: observe (X, Y), make prediction.
  e.g. predict a house's price (Y) from its size, location and age (X).

Terminology

ML uses new terms for old ideas:

- Estimation: Training
- Regressors: Features
- Parameters: Weights
- Supervised learning: observe (X, Y), make prediction.
  e.g. predict a house's price (Y) from its size, location and age (X).
- Unsupervised learning: observe X and clustering or structure discovery.
  e.g. group households into consumer types from their spending patterns (no Y).

Terminology

ML uses new terms for old ideas:

- Estimation: Training
- Regressors: Features
- Parameters: Weights
- Supervised learning: observe (X, Y), make prediction.
  e.g. predict a house's price (Y) from its size, location and age (X).
- Unsupervised learning: observe X and clustering or structure discovery.
  e.g. group households into consumer types from their spending patterns (no Y).
- Classification: ML term for discrete outcome models.
  e.g. predict whether a borrower defaults on a loan (yes/no) from income and credit history — in econometrics, a logit/probit.

Terminology

ML uses new terms for old ideas:

- Estimation: Training
- Regressors: Features
- Parameters: Weights
- Supervised learning: observe (X, Y), make prediction.
  e.g. predict a house's price (Y) from its size, location and age (X).
- Unsupervised learning: observe X and clustering or structure discovery.
  e.g. group households into consumer types from their spending patterns (no Y).
- Classification: ML term for discrete outcome models.
  e.g. predict whether a borrower defaults on a loan (yes/no) from income and credit history — in econometrics, a logit/probit.
- In Econometrics, the dimension of X is k; in Machine Learning it is p.

We start from a linear regression model:

$$Y_i = \alpha + \beta' X_i + \varepsilon_i, \qquad \varepsilon_i \mid X_i \sim \mathcal{N}(0, \sigma^2)$$

Equivalently, we model the conditional distribution of the outcome:

$$Y_i \mid X_i \sim \mathcal{N}(\alpha + \beta' X_i, \sigma^2)$$

e.g. Estimate parameters by **least squares**:

$$(\hat{\alpha}_{LS}, \hat{\beta}_{LS}) = \arg\min_{\alpha,\beta} \sum_{i=1}^{N} (Y_i - \alpha - \beta' X_i)^2$$

If the model is correct + some conditions:

- Estimator is **unbiased** and **BLUE** and has desirable **large-sample efficiency**

Prediction in Machine Learning

In **ML**, the focus is on **prediction** rather than estimation.

Predict a new outcome:

$$\hat{Y}_{N+1} = \hat{\alpha} + \hat{\beta}' X_{N+1}$$

Prediction in Machine Learning

In **ML**, the focus is on **prediction** rather than estimation.

Predict a new outcome:

$$\hat{Y}_{N+1} = \hat{\alpha} + \hat{\beta}' X_{N+1}$$

Minimize **out-of-sample loss**:

$$(Y_{N+1} - \hat{Y}_{N+1})^2$$

Prediction in Machine Learning

In **ML**, the focus is on **prediction** rather than estimation.

Predict a new outcome:

$$\hat{Y}_{N+1} = \hat{\alpha} + \hat{\beta}' X_{N+1}$$

Minimize **out-of-sample loss**:

$$(Y_{N+1} - \hat{Y}_{N+1})^2$$

- The estimators $(\hat{\alpha}, \hat{\beta})$ need **not** come from OLS.
- Any method that improves predictive accuracy — e.g., **regularization, trees, or boosting** — may be preferred.

(Cross)Validation

Econometrics (Validation):

- The form of the regression model, parametric or nonparametric, and the regressors are *given from the outside*, e.g., **economic theory**.
- Discussion of model selection is often in the form of **testing null hypotheses** concerning the validity of a particular model.

(Cross)Validation

Econometrics (Validation):

- The form of the regression model, parametric or nonparametric, and the regressors are *given from the outside*, e.g., **economic theory**.
- Discussion of model selection is often in the form of **testing null hypotheses** concerning the validity of a particular model.

Machine Learning (Cross-validation):

- Out-of-sample cross-validation can help guide such decisions.

(Cross)Validation

Econometrics (Validation):

- The form of the regression model, parametric or nonparametric, and the regressors are *given from the outside*, e.g., **economic theory**.
- Discussion of model selection is often in the form of **testing null hypotheses** concerning the validity of a particular model.

Machine Learning (Cross-validation):

- Out-of-sample cross-validation can help guide such decisions.

There are two components:

- **The goal is predictive power**, rather than estimation of a particular structural or causal parameter.
- The method uses **out-of-sample comparisons**, rather than in-sample goodness-of-fit measures.

Overfitting and Regularization

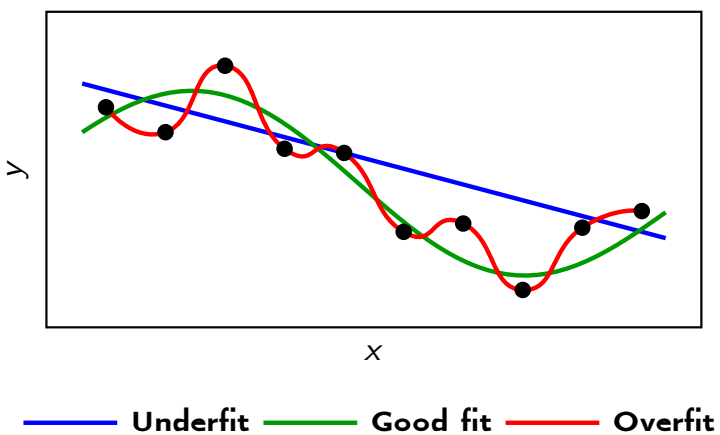
Overfitting (more of ML):

- Select flexible models that fit well, but not so well that out-of-sample prediction is compromised.
- Less emphasis on formal results, i.e., that particular methods are superior in large samples (asymptotically). Instead, methods are compared on specific data.

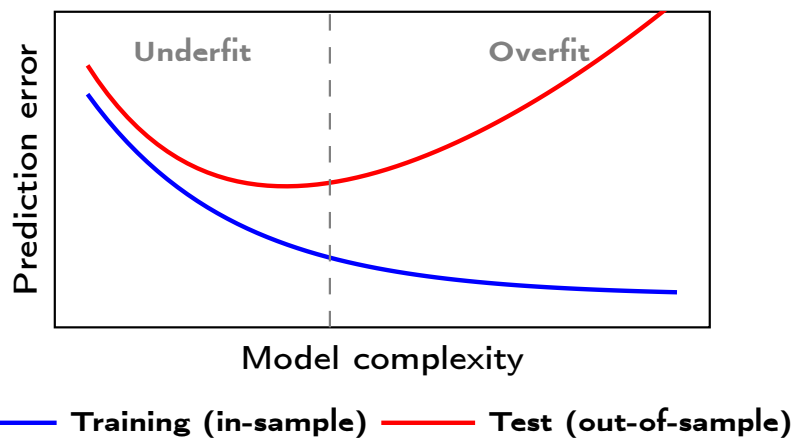
Overfitting (more of ML):

- Select flexible models that fit well, but not so well that out-of-sample prediction is compromised.
- Less emphasis on formal results, i.e., that particular methods are superior in large samples (asymptotically). Instead, methods are compared on specific data.

Fit to training data



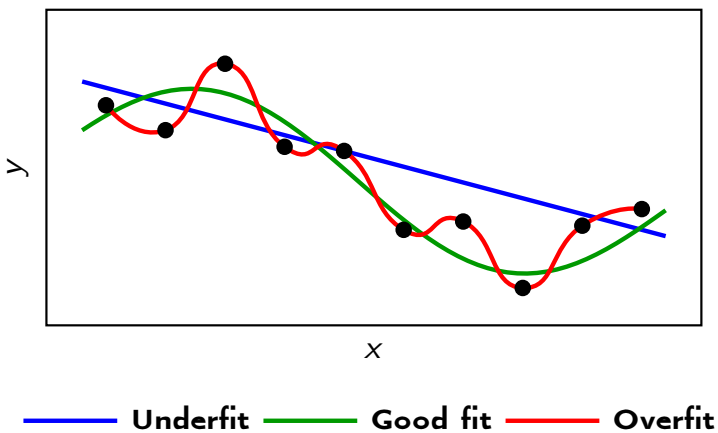
In- vs. out-of-sample error



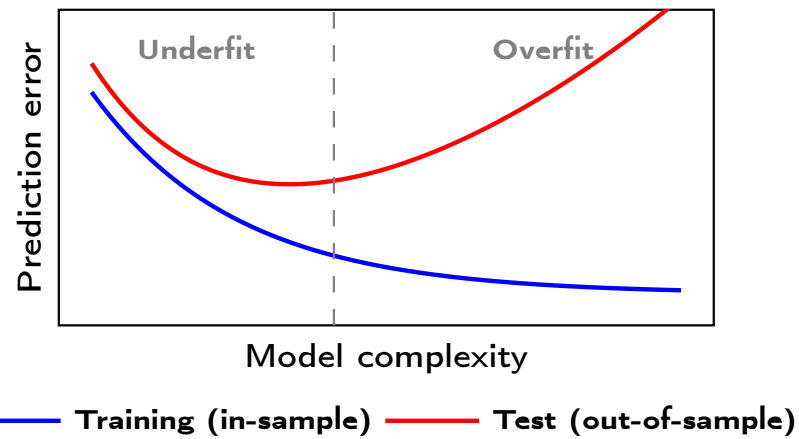
Overfitting (more of ML):

- Select flexible models that fit well, but not so well that out-of-sample prediction is compromised.
- Less emphasis on formal results, i.e., that particular methods are superior in large samples (asymptotically). Instead, methods are compared on specific data.

Fit to training data



In- vs. out-of-sample error



Regularization: add a **penalty for complexity** to the estimation problem, so the model fits the data well *without* overfitting.

Regularization

Idea: reward fit, but **penalize complexity**.

$$\text{Objective} = \text{Fit} \; - \; \text{Penalty}$$

Machine Learning: The strength of the penalty is chosen from the **data**: whatever predicts best out of sample.

Regularization

Idea: reward fit, but **penalize complexity**.

$$\text{Objective} = \text{Fit} \; - \; \text{Penalty}$$

Machine Learning: The strength of the penalty is chosen from the **data**: whatever predicts best out of sample.

Example:

$$\hat{\beta}_{Ridge} = \arg\min_{\beta} \; \sum_{i=1}^{N} (Y_i - \beta' X_i)^2 \; + \; \lambda \sum_{k=1}^{K} \beta_k^2$$

What λ (penalty parameter) does:

- $\lambda = 0$

Regularization

Idea: reward fit, but **penalize complexity**.

$$\text{Objective} = \text{Fit} \; - \; \text{Penalty}$$

Machine Learning: The strength of the penalty is chosen from the **data**: whatever predicts best out of sample.

Example:

$$\hat{\beta}_{Ridge} = \arg\min_{\beta} \; \sum_{i=1}^{N} (Y_i - \beta' X_i)^2 \; + \; \lambda \sum_{k=1}^{K} \beta_k^2$$

What λ (penalty parameter) does:

- $\lambda = 0 \Rightarrow$ no penalty $\Rightarrow$ **OLS**.

Regularization

Idea: reward fit, but **penalize complexity**.

$$\text{Objective} = \text{Fit} \; - \; \text{Penalty}$$

Machine Learning: The strength of the penalty is chosen from the **data**: whatever predicts best out of sample.

Example:

$$\hat{\beta}_{Ridge} = \arg\min_{\beta} \; \sum_{i=1}^{N} (Y_i - \beta' X_i)^2 \; + \; \lambda \sum_{k=1}^{K} \beta_k^2$$

What λ (penalty parameter) does:

- $\lambda = 0 \Rightarrow$ no penalty $\Rightarrow$ **OLS**.
- $\lambda \to \infty$

Regularization

Idea: reward fit, but **penalize complexity**.

$$\text{Objective} = \text{Fit} \; - \; \text{Penalty}$$

Machine Learning: The strength of the penalty is chosen from the **data**: whatever predicts best out of sample.

Example:

$$\hat{\beta}_{Ridge} = \arg\min_{\beta} \; \sum_{i=1}^{N} (Y_i - \beta' X_i)^2 \; + \; \lambda \sum_{k=1}^{K} \beta_k^2$$

What λ (penalty parameter) does:

- $\lambda = 0 \Rightarrow$ no penalty $\Rightarrow$ **OLS**.
- $\lambda \to \infty \Rightarrow$ all $\beta_k \to 0$.

Regularization

Idea: reward fit, but **penalize complexity**.

$$\text{Objective} = \text{Fit} \; - \; \text{Penalty}$$

Machine Learning: The strength of the penalty is chosen from the **data**: whatever predicts best out of sample.

Example:

$$\hat{\beta}_{Ridge} = \arg\min_{\beta} \; \sum_{i=1}^{N} (Y_i - \beta' X_i)^2 \; + \; \lambda \sum_{k=1}^{K} \beta_k^2$$

What λ (penalty parameter) does:

- $\lambda = 0 \Rightarrow$ no penalty $\Rightarrow$ **OLS**.
- $\lambda \to \infty \Rightarrow$ all $\beta_k \to 0$.
- In between:

Regularization

Idea: reward fit, but **penalize complexity**.

$$\text{Objective} = \text{Fit} \; - \; \text{Penalty}$$

Machine Learning: The strength of the penalty is chosen from the **data**: whatever predicts best out of sample.

Example:

$$\hat{\beta}_{Ridge} = \arg\min_{\beta} \; \sum_{i=1}^{N} (Y_i - \beta' X_i)^2 \; + \; \lambda \sum_{k=1}^{K} \beta_k^2$$

What λ (penalty parameter) does:

- $\lambda = 0 \Rightarrow$ no penalty $\Rightarrow$ **OLS**.
- $\lambda \to \infty \Rightarrow$ all $\beta_k \to 0$.
- In between: coefficients are **shrunk** toward 0 $\Rightarrow$ more stable predictions, less overfitting.

Regularization

Idea: reward fit, but **penalize complexity**.

$$\text{Objective} = \text{Fit} \; - \; \text{Penalty}$$

Machine Learning: The strength of the penalty is chosen from the **data**: whatever predicts best out of sample.

Example:

$$\hat{\beta}_{Ridge} = \arg\min_{\beta} \; \sum_{i=1}^{N} (Y_i - \beta' X_i)^2 \; + \; \lambda \sum_{k=1}^{K} \beta_k^2$$

What λ (penalty parameter) does:

- $\lambda = 0 \Rightarrow$ no penalty $\Rightarrow$ **OLS**.
- $\lambda \to \infty \Rightarrow$ all $\beta_k \to 0$.
- In between: coefficients are **shrunk** toward 0 $\Rightarrow$ more stable predictions, less overfitting.

How to choose λ:

- ML: by **cross-validation** — the λ that predicts best out of sample.

Regularization

- Econometrics motivation is to reduce the degree of collinearity among the regressors.
- ML motivation is regularization of high-dimensional problems (too many regressors).
- Traditional parametric asymptotic theory assumes that p is fixed as $n \to \infty$, implying that p is much smaller than n.
- A **high-dimensional setup** is used to describe the context where $p \to \infty$, including when $p > n$.

Outline

- Introduction
- Ridge Regression
- Lasso Estimator
- Elastic Net

High Dimensional Problem

Given $\beta_{ols} = (X'X)^{-1}X'Y$:

- When $p > n$:

High Dimensional Problem

Given $\beta_{ols} = (X'X)^{-1}X'Y$:

- When $p > n$: the estimator β_{ols} is not defined since $X'X$ has deficient rank: more unknowns than equations.

High Dimensional Problem

Given $\beta_{ols} = (X'X)^{-1}X'Y$:

- When $p > n$: the estimator β_{ols} is not defined since $X'X$ has deficient rank: more unknowns than equations.
- If $p < n$ but p is large:

High Dimensional Problem

Given $\beta_{ols} = (X'X)^{-1}X'Y$:

- When $p > n$: the estimator β_{ols} is not defined since $X'X$ has deficient rank: more unknowns than equations.
- If $p < n$ but p is large: we can invert $X'X$ but it is ill-conditioned or nearly singular — multicollinearity near perfect.

High Dimensional Problem

Given $\beta_{ols} = (X'X)^{-1}X'Y$:

- When $p > n$: the estimator β_{ols} is not defined since $X'X$ has deficient rank: more unknowns than equations.
- If $p < n$ but p is large: we can invert $X'X$ but it is ill-conditioned or nearly singular — multicollinearity near perfect.
- The eigenvalues of $X'X$ tell us how much independent information the data contain along different linear combinations.

High Dimensional Problem

Given $\beta_{ols} = (X'X)^{-1}X'Y$:

- When $p > n$: the estimator β_{ols} is not defined since $X'X$ has deficient rank: more unknowns than equations.
- If $p < n$ but p is large: we can invert $X'X$ but it is ill-conditioned or nearly singular — multicollinearity near perfect.
- The eigenvalues of $X'X$ tell us how much independent information the data contain along different linear combinations.
- Small eigenvalues of $X'X$ arise because predictors/features convey nearly redundant information.

High Dimensional Problem: A Simple Example

Case 1: $p > n$ $(n = 1,\ p = 2)$

$$X = \begin{pmatrix} 1 & 2 \end{pmatrix}, \quad X'X = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$$

High Dimensional Problem: A Simple Example

Case 1: $p > n$ ($n = 1$, $p = 2$)

$$X = \begin{pmatrix} 1 & 2 \end{pmatrix}, \quad X'X = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$$

- $\det(X'X) = 4 - 4 = 0$, rank $= 1 < 2$.
- Eigenvalues: 5 and 0 $\Rightarrow$ no inverse.
- With $Y = 3$: one equation, two unknowns: $\beta_{\mathbf{1}} + 2\beta_{\mathbf{2}} = 3$.
- $(1, 1)$, $(3, 0)$, $(-1, 2)$, ... all fit *perfectly*.

High Dimensional Problem: A Simple Example

Case 1: $p > n$ $(n = 1,\ p = 2)$

$$X = \begin{pmatrix} 1 & 2 \end{pmatrix}, \quad X'X = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$$

- $\det(X'X) = 4 - 4 = 0$, rank $= 1 < 2$.
- Eigenvalues: 5 and 0 $\Rightarrow$ no inverse.
- With $Y = 3$: one equation, two unknowns: $\beta_{\mathbf{1}} + 2\beta_{\mathbf{2}} = 3$.
- $(1, 1)$, $(3, 0)$, $(-1, 2)$, ... all fit *perfectly*.

Case 2: near collinearity $(n = p = 2)$

$$X = \begin{pmatrix} 1 & 1 \\ 1 & 1.01 \end{pmatrix}, \quad X'X = \begin{pmatrix} 2 & 2.01 \\ 2.01 & 2.0201 \end{pmatrix}$$

High Dimensional Problem: A Simple Example

Case 1: $p > n$ $(n = 1,\ p = 2)$

$$X = \begin{pmatrix} 1 & 2 \end{pmatrix}, \quad X'X = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$$

- $\det(X'X) = 4 - 4 = 0$, rank $= 1 < 2$.
- Eigenvalues: 5 and 0 $\Rightarrow$ no inverse.
- With $Y = 3$: one equation, two unknowns: $\beta_{\mathbf{1}} + 2\beta_{\mathbf{2}} = 3$.
- $(1, 1)$, $(3, 0)$, $(-1, 2)$, ... all fit *perfectly*.

Case 2: near collinearity $(n = p = 2)$

$$X = \begin{pmatrix} 1 & 1 \\ 1 & 1.01 \end{pmatrix}, \quad X'X = \begin{pmatrix} 2 & 2.01 \\ 2.01 & 2.0201 \end{pmatrix}$$

- $x_{\mathbf{2}} \approx x_{\mathbf{1}}$; $\det(X'X) = 0.0001$.
- Eigenvalues: ≈ 4.02 and ≈ 0.000025.
- $Y = (2,\ 2.01)' \Rightarrow \hat{\beta} = (1,\ 1)'$
- $Y = (2,\ 2.02)' \Rightarrow \hat{\beta} = (0,\ 2)'$

High Dimensional Problem

Remember $(X'X)^{-1}$ contains covariance in predictors/features.

- This can be expressed as terms $1/\text{eigenvalue}$ that explode with small variations in data.
- Many of the variables may be low-information, but it is difficult to know a priori which.

High Dimensional Problem

Remember $(X'X)^{-1}$ contains covariance in predictors/features.

- This can be expressed as terms 1/eigenvalue that explode with small variations in data.
- Many of the variables may be low-information, but it is difficult to know a priori which.

Consequently, **we turn to estimation methods other than least squares.**

- Say ridge regression, Lasso, elastic net, regression trees, and random forests.

Ridge Regression: Addressing Ill-Conditioning

- **Problem with OLS:** When regressors are highly correlated, $X'X$ can be ill-conditioned (nearly singular).
  - Small changes in the data may produce large changes in $\hat{\beta}_{OLS}$.
  - If $p > n$, $X'X$ is singular and cannot be inverted.

Ridge Regression: Addressing Ill-Conditioning

- **Problem with OLS:** When regressors are highly correlated, $X'X$ can be ill-conditioned (nearly singular).
  - Small changes in the data may produce large changes in $\hat{\beta}_{OLS}$.
  - If $p > n$, $X'X$ is singular and cannot be inverted.

- **Ridge solution:** Add a positive constant λ to the diagonal of $X'X$:

$$\boxed{\hat{\beta}_{Ridge} = (X'X + \lambda I_p)^{-1} X'Y}$$

Ridge Regression: Addressing Ill-Conditioning

- **Problem with OLS:** When regressors are highly correlated, $X'X$ can be ill-conditioned (nearly singular).
  - Small changes in the data may produce large changes in $\hat{\beta}_{OLS}$.
  - If $p > n$, $X'X$ is singular and cannot be inverted.

- **Ridge solution:** Add a positive constant λ to the diagonal of $X'X$:

$$\boxed{\hat{\beta}_{Ridge} = (X'X + \lambda I_p)^{-1} X'Y}$$

- **Why does this help?**

Ridge Regression: Addressing Ill-Conditioning

- **Problem with OLS:** When regressors are highly correlated, $X'X$ can be ill-conditioned (nearly singular).
  - Small changes in the data may produce large changes in $\hat{\beta}_{OLS}$.
  - If $p > n$, $X'X$ is singular and cannot be inverted.

- **Ridge solution:** Add a positive constant λ to the diagonal of $X'X$:

$$\boxed{\hat{\beta}_{Ridge} = (X'X + \lambda I_p)^{-1} X'Y}$$

- **Why does this help?**
  - $\lambda > 0$ makes $X'X + \lambda I_p$ invertible, even when $p > n$.
  - Increases small eigenvalues, improving numerical stability.
  - Larger λ means more shrinkage toward zero.

Ridge Regression: Addressing Ill-Conditioning

- **Problem with OLS:** When regressors are highly correlated, $X'X$ can be ill-conditioned (nearly singular).
  - Small changes in the data may produce large changes in $\hat{\beta}_{OLS}$.
  - If $p > n$, $X'X$ is singular and cannot be inverted.

- **Ridge solution:** Add a positive constant λ to the diagonal of $X'X$:

$$\boxed{\hat{\beta}_{Ridge} = (X'X + \lambda I_p)^{-1} X'Y}$$

- **Why does this help?**
  - $\lambda > 0$ makes $X'X + \lambda I_p$ invertible, even when $p > n$.
  - Increases small eigenvalues, improving numerical stability.
  - Larger λ means more shrinkage toward zero.

- **Trade-off:** Ridge introduces bias but can reduce variance and improve out-of-sample predictions.

Ridge Regression: Addressing Ill-Conditioning

- **Problem with OLS:** When regressors are highly correlated, $X'X$ can be ill-conditioned (nearly singular).
  - Small changes in the data may produce large changes in $\hat{\beta}_{OLS}$.
  - If $p > n$, $X'X$ is singular and cannot be inverted.

- **Ridge solution:** Add a positive constant λ to the diagonal of $X'X$:

$$\boxed{\hat{\beta}_{Ridge} = (X'X + \lambda I_p)^{-1} X'Y}$$

- **Why does this help?**
  - $\lambda > 0$ makes $X'X + \lambda I_p$ invertible, even when $p > n$.
  - Increases small eigenvalues, improving numerical stability.
  - Larger λ means more shrinkage toward zero.

- **Trade-off:** Ridge introduces bias but can reduce variance and improve out-of-sample predictions.
- **Ridge and Machine Learning:** Ridge still uses a linear prediction function ($\hat{Y} = X\hat{\beta}$), but does not require the true population relationship $E[Y \mid X]$ to be linear. The **goal is to minimize out-of-sample prediction error rather than necessarily estimate a correctly specified population model.**

Ridge Regression: Spectral Decomposition

Spectral decomposition:

$$X'X = H'DH$$

- H is an orthonormal matrix whose rows are eigenvectors of $X'X$.
- Orthonormal means $H'H = HH' = I$: H rotates or reflects the system without rescaling.
- $D = \text{diag}(r_1, \ldots, r_p)$ contains the eigenvalues of $X'X$.

Ridge Regression: Spectral Decomposition

Spectral decomposition:

$$X'X = H'DH$$

- H is an orthonormal matrix whose rows are eigenvectors of $X'X$.
- Orthonormal means $H'H = HH' = I$: H rotates or reflects the system without rescaling.
- $D = \text{diag}(r_1, \ldots, r_p)$ contains the eigenvalues of $X'X$.

Example with three regressors ($p = 3$):

$$\underbrace{X'X}_{3\times 3} = \underbrace{\begin{pmatrix} h_{11} & h_{21} & h_{31} \\ h_{12} & h_{22} & h_{32} \\ h_{13} & h_{23} & h_{33} \end{pmatrix}}_{H'} \underbrace{\begin{pmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{pmatrix}}_{D} \underbrace{\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}}_{H}$$

the coefficient space (β-space).

- **Eigenvalues:** Measure the amount of variation/information along each direction.
- **Ill-conditioning:** Occurs when some eigenvalues are very small relative to others.

Ridge Regression

Set $\Lambda = \lambda I_p$:

$$X'X + \lambda I_p = H'DH + \lambda H'H = H'(D + \Lambda)H$$

Which has strictly positive eigenvalues $r_j + \lambda > 0$.

- $\lambda = 0$: gives the OLS estimator.
- When inverting: $H'(D + \Lambda)^{-1}H$, think of it as $1/(D + \lambda)$.
- If eigenvalues (in D) are small and $\lambda = 0$: i.e. high correlation between regressors, β "explodes."
- Ridge adds λ to every eigenvalue, stabilizing the inverse while keeping the same eigenvectors.

Ridge Regression: Penalization and the Estimator

Ridge minimizes the sum of squared errors plus a penalty:

$$\min_{\beta} \left\{ (Y - X\beta)'(Y - X\beta) + \lambda \beta'\beta \right\}$$

Ridge Regression: Penalization and the Estimator

Ridge minimizes the sum of squared errors plus a penalty:

$$\min_{\beta} \left\{ (Y - X\beta)'(Y - X\beta) + \lambda \beta'\beta \right\}$$

First-order condition:

$$-2X'Y + 2X'X\beta + 2\lambda\beta = 0$$

Ridge Regression: Penalization and the Estimator

Ridge minimizes the sum of squared errors plus a penalty:

$$\min_{\beta} \left\{ (Y - X\beta)'(Y - X\beta) + \lambda \beta'\beta \right\}$$

First-order condition:

$$-2X'Y + 2X'X\beta + 2\lambda\beta = 0$$

Rearranging and solving for β:

$$(X'X + \lambda I_p)\beta = X'Y$$

$$\boxed{\hat{\beta}_{Ridge} = (X'X + \lambda I_p)^{-1} X'Y}$$

Ridge Regression: Penalization and the Estimator

Ridge minimizes the sum of squared errors plus a penalty:

$$\min_{\beta} \left\{ (Y - X\beta)'(Y - X\beta) + \lambda \beta'\beta \right\}$$

First-order condition:

$$-2X'Y + 2X'X\beta + 2\lambda\beta = 0$$

Rearranging and solving for β:

$$(X'X + \lambda I_p)\beta = X'Y$$

$$\boxed{\hat{\beta}_{Ridge} = (X'X + \lambda I_p)^{-1} X'Y}$$

Key insight:

- Adding an L_2 penalty to the SSE is equivalent to adding λI_p to $X'X$.
- This improves numerical stability and shrinks coefficients toward zero.
- $\lambda = 0$ recovers OLS when $X'X$ is invertible.

Ridge Regression (Regularization)

Hence, the second motivation for using Ridge is:

- When $X'X$ is ill-conditioned its inverse is *ill-posed* (i.e. numerically unstable): nearly singular or $p \geq n$ (not unique).
- Techniques to deal with ill-posed estimators are called *regularization*.
- This could be made through *Penalization*.

Ridge Regression (Regularization)

Consider:

$$SSE_2(\beta, \lambda) = (Y - X\beta)'(Y - X\beta) + \lambda\beta'\beta = \|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2.$$

The minimizer of SSE_2 is a regularized least squares estimator:

$$SSE_2(\beta, \lambda) = \underbrace{\|Y - X\beta\|_2^2}_{\text{Fit term}} + \underbrace{\lambda\|\beta\|_2^2}_{\text{Penalty term}}$$

Ridge Regression (Regularization)

Consider:

$$SSE_2(\beta, \lambda) = (Y - X\beta)'(Y - X\beta) + \lambda\beta'\beta = \|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2.$$

The minimizer of SSE_2 is a regularized least squares estimator:

$$SSE_2(\beta, \lambda) = \underbrace{\|Y - X\beta\|_2^2}_{\text{Fit term}} + \underbrace{\lambda\|\beta\|_2^2}_{\text{Penalty term}}$$

- The **fit term** rewards accurate predictions.

Ridge Regression (Regularization)

Consider:

$$SSE_2(\beta, \lambda) = (Y - X\beta)'(Y - X\beta) + \lambda\beta'\beta = \|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2.$$

The minimizer of SSE_2 is a regularized least squares estimator:

$$SSE_2(\beta, \lambda) = \underbrace{\|Y - X\beta\|_2^2}_{\text{Fit term}} + \underbrace{\lambda\|\beta\|_2^2}_{\text{Penalty term}}$$

- The **fit term** rewards accurate predictions.
- The **penalty term** discourages large coefficients.

Ridge Regression (Regularization)

Consider:

$$SSE_2(\beta, \lambda) = (Y - X\beta)'(Y - X\beta) + \lambda\beta'\beta = \|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2.$$

The minimizer of SSE_2 is a regularized least squares estimator:

$$SSE_2(\beta, \lambda) = \underbrace{\|Y - X\beta\|_2^2}_{\text{Fit term}} + \underbrace{\lambda\|\beta\|_2^2}_{\text{Penalty term}}$$

- The **fit term** rewards accurate predictions.
- The **penalty term** discourages large coefficients.
- Penalizing large coefficient vectors avoids them being too large and erratic.

Ridge Regression (Regularization)

Consider:

$$SSE_2(\beta, \lambda) = (Y - X\beta)'(Y - X\beta) + \lambda\beta'\beta = \|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2.$$

The minimizer of SSE_2 is a regularized least squares estimator:

$$SSE_2(\beta, \lambda) = \underbrace{\|Y - X\beta\|_2^2}_{\text{Fit term}} + \underbrace{\lambda\|\beta\|_2^2}_{\text{Penalty term}}$$

- The **fit term** rewards accurate predictions.
- The **penalty term** discourages large coefficients.
- Penalizing large coefficient vectors avoids them being too large and erratic.
- So Ridge avoids β estimators from exploding and regularizes them towards zero.

Numeric Example

Let $y = 2$, $x = 1$, and $\lambda = 1$:

β	Fit term $(y - x\beta)^2$	Penalty term $\lambda\beta^2$	Total $SSE_2(\beta, \lambda)$
0	4	0	4
1	1	1	2 (minimum)
2	0	4	4
3	1	9	10

Numeric Example

Let $y = 2$, $x = 1$, and $\lambda = 1$:

β	Fit term $(y - x\beta)^2$	Penalty term $\lambda\beta^2$	Total $SSE_2(\beta, \lambda)$
0	4	0	4
1	1	1	2 (minimum)
2	0	4	4
3	1	9	10

Question: What are $\hat{\beta}_{OLS}$ and $\hat{\beta}_{Ridge}$?

Numeric Example

Let $y = 2$, $x = 1$, and $\lambda = 1$:

β	Fit term $(y - x\beta)^2$	Penalty term $\lambda\beta^2$	Total $SSE_2(\beta, \lambda)$
0	4	0	4
1	1	1	2 (minimum)
2	0	4	4
3	1	9	10

Question: What are $\hat{\beta}_{OLS}$ and $\hat{\beta}_{Ridge}$?

$$\hat{\beta}_{OLS} = 2 \qquad \hat{\beta}_{Ridge} = 1$$

Numeric Example

Let $y = 2$, $x = 1$, and $\lambda = 1$:

β	Fit term $(y - x\beta)^2$	Penalty term $\lambda\beta^2$	Total $SSE_2(\beta, \lambda)$
0	4	0	4
1	1	1	2 (minimum)
2	0	4	4
3	1	9	10

Question: What are $\hat{\beta}_{OLS}$ and $\hat{\beta}_{Ridge}$?

$$\boxed{\hat{\beta}_{OLS} = 2 \qquad \hat{\beta}_{Ridge} = 1}$$

- Ridge **shrinks** coefficients toward zero.
- Large λ: stronger "pull" toward zero.
- Small λ: weaker penalty, closer to OLS.

Ridge Regression (Regularization)

You can either think of Ridge regression as penalization (λ) or as a constraint (τ):

$$\min_{\beta} \|Y - X\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_2^2 \leq \tau$$

- Here, $\tau \geq 0$ is the maximum allowed "size" of β.
- You *control* the allowed size of coefficients. Hence, the larger τ, the smaller λ, and vice versa.

Ridge Regression (Regularization)

From constrained optimization to penalization:

- Recall that Ridge can be written as:

$$\min_{\beta} \ \|Y - X\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_2^2 \leq \tau$$

- To solve this constrained problem, we use the **Lagrangian method**: introduce a multiplier $\lambda \geq 0$ associated with the constraint.
- The Lagrangian function becomes:

$$\mathcal{L}(\beta, \lambda) = (Y - X\beta)'(Y - X\beta) + \lambda(\beta'\beta - \tau)$$

Ridge Regression (Regularization)

From constrained optimization to penalization:

- Recall that Ridge can be written as:

$$\min_{\beta} \ \|Y - X\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_2^2 \leq \tau$$

- To solve this constrained problem, we use the **Lagrangian method**: introduce a multiplier $\lambda \geq 0$ associated with the constraint.
- The Lagrangian function becomes:

$$\mathcal{L}(\beta, \lambda) = (Y - X\beta)'(Y - X\beta) + \lambda(\beta'\beta - \tau)$$

First-order condition with respect to β:

$$\frac{\partial \mathcal{L}}{\partial \beta} = -2X'(Y - X\beta) + 2\lambda\beta = 0$$

Ridge Regression (Regularization)

From constrained optimization to penalization:

- Recall that Ridge can be written as:

$$\min_{\beta} \ \|Y - X\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_2^2 \leq \tau$$

- To solve this constrained problem, we use the **Lagrangian method**: introduce a multiplier $\lambda \geq 0$ associated with the constraint.
- The Lagrangian function becomes:

$$\mathcal{L}(\beta, \lambda) = (Y - X\beta)'(Y - X\beta) + \lambda(\beta'\beta - \tau)$$

First-order condition with respect to β:

$$\frac{\partial \mathcal{L}}{\partial \beta} = -2X'(Y - X\beta) + 2\lambda\beta = 0$$

$$\boxed{\hat{\beta}_{Ridge} = (X'X + \lambda I_p)^{-1} X'Y}$$

- For a given τ, the corresponding λ is determined by the constraint and complementary slackness.
- In ML, λ is typically selected using **cross-validation**.

Ridge Regression: Dual Minimization

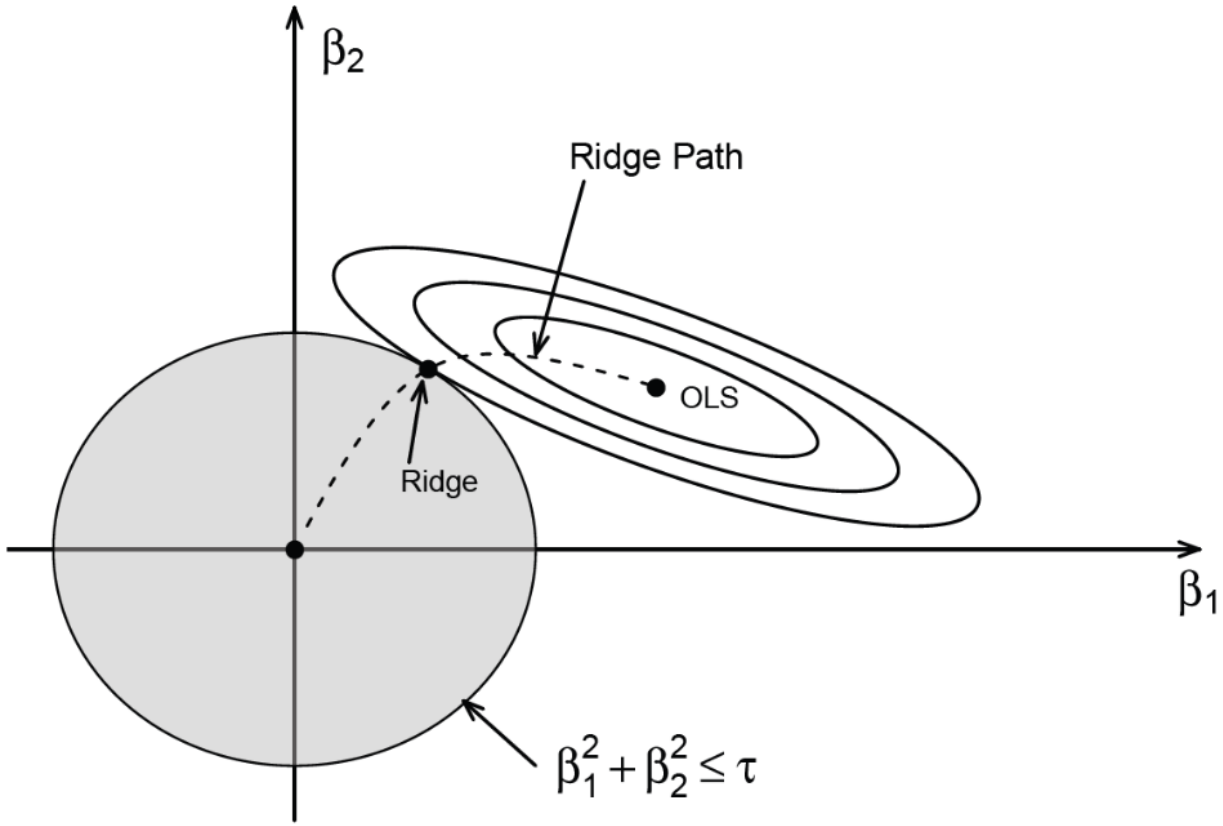


Figure 29.1: Ridge Regression Dual Minimization Solution

Ridge "path" is given by varying sizes of λ. The sphere's contour is given by τ.

Ridge Regression in Practice: Cross-Validation

How do we choose the shrinkage parameter τ?

- One approach uses prior information (Bayesian).
- Another is **cross-validation (CV)**: choose τ to minimize out-of-sample prediction error.

Ridge Regression in Practice: Cross-Validation

How do we choose the shrinkage parameter τ?

- One approach uses prior information (Bayesian).
- Another is **cross-validation (CV)**: choose τ to minimize out-of-sample prediction error.

Steps of K-fold cross-validation:

1. Split the data into K approximately equal folds.

Ridge Regression in Practice: Cross-Validation

How do we choose the shrinkage parameter τ?

- One approach uses prior information (Bayesian).
- Another is **cross-validation (CV)**: choose τ to minimize out-of-sample prediction error.

Steps of K-fold cross-validation:

1. Split the data into K approximately equal folds.
2. Choose a grid of candidate τ values (smaller τ means stronger shrinkage).

Ridge Regression in Practice: Cross-Validation

How do we choose the shrinkage parameter τ?

- One approach uses prior information (Bayesian).
- Another is **cross-validation (CV)**: choose τ to minimize out-of-sample prediction error.

Steps of K-fold cross-validation:

1. Split the data into K approximately equal folds.
2. Choose a grid of candidate τ values (smaller τ means stronger shrinkage).
3. For each τ:
   - Fit Ridge using $K - 1$ folds:

Ridge Regression in Practice: Cross-Validation

How do we choose the shrinkage parameter τ?

- One approach uses prior information (Bayesian).
- Another is **cross-validation (CV)**: choose τ to minimize out-of-sample prediction error.

Steps of K-fold cross-validation:

1. Split the data into K approximately equal folds.
2. Choose a grid of candidate τ values (smaller τ means stronger shrinkage).
3. For each τ:
   - Fit Ridge using $K-1$ folds: $\hat{\beta}^{(-k)}(\tau)$
   - Predict outcomes in the left-out fold k:

Ridge Regression in Practice: Cross-Validation

How do we choose the shrinkage parameter τ?

- One approach uses prior information (Bayesian).
- Another is **cross-validation (CV)**: choose τ to minimize out-of-sample prediction error.

Steps of K-fold cross-validation:

1. Split the data into K approximately equal folds.
2. Choose a grid of candidate τ values (smaller τ means stronger shrinkage).
3. For each τ:
   - Fit Ridge using $K-1$ folds: $\hat{\beta}^{(-k)}(\tau)$
   - Predict outcomes in the left-out fold k: $\hat{Y}_i^{(-k)}(\tau) = X_i' \hat{\beta}^{(-k)}(\tau), \quad i \in F_k$
   - Compute the mean squared prediction error:

Ridge Regression in Practice: Cross-Validation

How do we choose the shrinkage parameter τ?

- One approach uses prior information (Bayesian).
- Another is **cross-validation (CV)**: choose τ to minimize out-of-sample prediction error.

Steps of K-fold cross-validation:

1. Split the data into K approximately equal folds.
2. Choose a grid of candidate τ values (smaller τ means stronger shrinkage).
3. For each τ:
   - Fit Ridge using $K-1$ folds: $\hat{\beta}^{(-k)}(\tau)$
   - Predict outcomes in the left-out fold k: $\hat{Y}_i^{(-k)}(\tau) = X_i' \hat{\beta}^{(-k)}(\tau), \quad i \in F_k$
   - Compute the mean squared prediction error:

$$MSE_k(\tau) = \frac{1}{n_k} \sum_{i \in F_k} (Y_i - \hat{Y}_i^{(-k)}(\tau))^2$$

   - Repeat for all K folds.

Ridge Regression in Practice: Cross-Validation

How do we choose the shrinkage parameter τ?

- One approach uses prior information (Bayesian).
- Another is **cross-validation (CV)**: choose τ to minimize out-of-sample prediction error.

Steps of K-fold cross-validation:

1. Split the data into K approximately equal folds.
2. Choose a grid of candidate τ values (smaller τ means stronger shrinkage).
3. For each τ:
   - Fit Ridge using $K-1$ folds: $\hat{\beta}^{(-k)}(\tau)$
   - Predict outcomes in the left-out fold k: $\hat{Y}_i^{(-k)}(\tau) = X_i' \hat{\beta}^{(-k)}(\tau), \quad i \in F_k$
   - Compute the mean squared prediction error:

$$MSE_k(\tau) = \frac{1}{n_k} \sum_{i \in F_k} (Y_i - \hat{Y}_i^{(-k)}(\tau))^2$$

   - Repeat for all K folds.
4. Average prediction errors across folds:

$$CV(\tau) = \frac{1}{K} \sum_{k=1}^{K} MSE_k(\tau)$$

Ridge Regression in Practice: Cross-Validation

How do we choose the shrinkage parameter τ?

- One approach uses prior information (Bayesian).
- Another is **cross-validation (CV)**: choose τ to minimize out-of-sample prediction error.

Steps of K-fold cross-validation:

1. Split the data into K approximately equal folds.
2. Choose a grid of candidate τ values (smaller τ means stronger shrinkage).
3. For each τ:
   - Fit Ridge using $K-1$ folds: $\hat{\beta}^{(-k)}(\tau)$
   - Predict outcomes in the left-out fold k: $\hat{Y}_i^{(-k)}(\tau) = X_i' \hat{\beta}^{(-k)}(\tau), \quad i \in F_k$
   - Compute the mean squared prediction error:

$$MSE_k(\tau) = \frac{1}{n_k} \sum_{i \in F_k} (Y_i - \hat{Y}_i^{(-k)}(\tau))^2$$

   - Repeat for all K folds.
4. Average prediction errors across folds:

$$CV(\tau) = \frac{1}{K} \sum_{k=1}^{K} MSE_k(\tau)$$

5. Select τ that minimizes $CV(\tau)$.

Goal: Balance bias (underfitting) and variance (overfitting) to improve predictions.

The most common method is cross-validation of λ.
The leave-one-out (LOO) ridge estimator, prediction errors, and CV criterion are:

$$\hat{\beta}_{-i}(\lambda) = \left(\sum_{j \neq i} X_j X_j' + \Lambda \right)^{-1} \left(\sum_{j \neq i} X_j Y_j \right),$$

$$\tilde{e}_i(\lambda) = Y_i - X_i' \hat{\beta}_{-i}(\lambda),$$

$$CV(\lambda) = \sum_{i=1}^{n} \tilde{e}_i(\lambda)^2.$$

Choose the $\hat{\lambda}$ that minimizes the cross-validation error $CV(\lambda)$ and use it to calculate the cross-validation ridge.

Code

Ridge Regression: Covariance Matrix

Under random sampling, the covariance matrix of the Ridge estimator is:

$$\text{Var}[\hat{\beta}_{ridge} \mid X] = (X'X + \lambda I_p)^{-1}(X'DX)(X'X + \lambda I_p)^{-1}$$

Where:

$$D = \text{diag}\{\sigma^2(X_1), \ldots, \sigma^2(X_n)\} \quad \text{and} \quad \sigma^2(x) = \mathbb{E}[e^2 \mid X = x].$$

Ridge Regression: Covariance Matrix

- When errors have constant variance (homoskedasticity), $D = \sigma^2 I_n$, so:

$$\text{Var}[\hat{\beta}_{ridge} \mid X] = \sigma^2 (X'X + \lambda I_p)^{-1} X'X (X'X + \lambda I_p)^{-1}.$$

- Under clustering or serial correlation, the middle term $X'DX$ is modified accordingly.
- This expression shows how λ stabilizes the inversion of $X'X$, **reducing the variance of the estimated coefficients**.

Overall, $\hat{\beta}_{ridge}$ plus $\text{Var}[\hat{\beta}_{ridge}|X]$ give us the nature of the **bias–variance trade-off of the Ridge estimator**.

Ridge Regression: Covariance Matrix

Warning:

- Standard error interpretation in *Ridge Regression* is non-standard. β's are biased by construction.
- Confidence intervals will have deficient coverage.
- Ridge is generally employed in prediction with no standard errors.
- Note: Ridge regression will include all p predictors in the final model.

Now **in Stata:** Download and perform OLS and Ridge Regression: Code

Outline

- Introduction
- Ridge Regression
- Lasso Estimator
- Elastic Net

The Lasso Estimator

An alternative to Ridge uses the L_1 norm penalty ($\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$, the sum of the absolute values of the coefficients):

- Known as the **Lasso** (*Least Absolute Shrinkage and Selection Operator*).

Objective Function:

- The least squares criterion with an L_1 (absolute) penalty is:

The Lasso Estimator

An alternative to Ridge uses the L_1 norm penalty ($\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$, the sum of the absolute values of the coefficients):

- Known as the **Lasso** (*Least Absolute Shrinkage and Selection Operator*).

Objective Function:

- The least squares criterion with an L_1 (absolute) penalty is:

$$SSE_1(\beta, \lambda) = (Y - X\beta)'(Y - X\beta) + \lambda \sum_{j=1}^{p} |\beta_j| = \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

The Lasso estimator is the minimizer of this objective:

$$\hat{\beta}_{Lasso} = \arg\min_{\beta} SSE_1(\beta, \lambda)$$

The Lasso Estimator

- For $\lambda > 0$, the Lasso estimator is **well-defined even when** $p > n$.
- The solution generally must be found **numerically**.
- The L_1 penalty induces **sparsity**: e.g. some coefficients are set exactly to zero, combining shrinkage and variable selection.

The Lasso Estimator

The Lasso minimization problem uses the dual constrained minimization problem:

$$\hat{\beta}_{Lasso} = \arg \min_{\|\beta\|_1 \leq \tau} SSE_1(\beta)$$

The Lasso Estimator

The Lasso minimization problem uses the dual constrained minimization problem:

$$\hat{\beta}_{Lasso} = \arg \min_{\|\beta\|_1 \leq \tau} SSE_1(\beta)$$

Observe that the constrained minimization has the Lagrangian:

The Lasso Estimator

The Lasso minimization problem uses the dual constrained minimization problem:

$$\hat{\beta}_{Lasso} = \arg \min_{\|\beta\|_1 \leq \tau} SSE_1(\beta)$$

Observe that the constrained minimization has the Lagrangian:

$$\min_{\beta}(Y - X\beta)'(Y - X\beta) + \lambda \left(\sum_{j=1}^{p} |\beta_j| - \tau \right)$$

The Lasso Estimator

The Lasso minimization problem uses the dual constrained minimization problem:

$$\hat{\beta}_{Lasso} = \arg \min_{\|\beta\|_1 \leq \tau} SSE_1(\beta)$$

Observe that the constrained minimization has the Lagrangian:

$$\min_{\beta}(Y - X\beta)'(Y - X\beta) + \lambda \left(\sum_{j=1}^{p} |\beta_j| - \tau \right)$$

Which has first-order conditions:

$$-2X_j'(Y - X\beta) + \lambda \, \text{sgn}(\beta_j) = 0.$$

$\text{sgn}(\beta_j)$ is the sign function, equal to $+1$ if $\beta_j > 0$, -1 if $\beta_j < 0$, and 0 if $\beta_j = 0$.

This is the same as those for minimization of the LASSO penalized criterion. Thus, for every value of λ, there exists a τ that yields $\hat{\beta}$.

Lasso: Dual Minimization

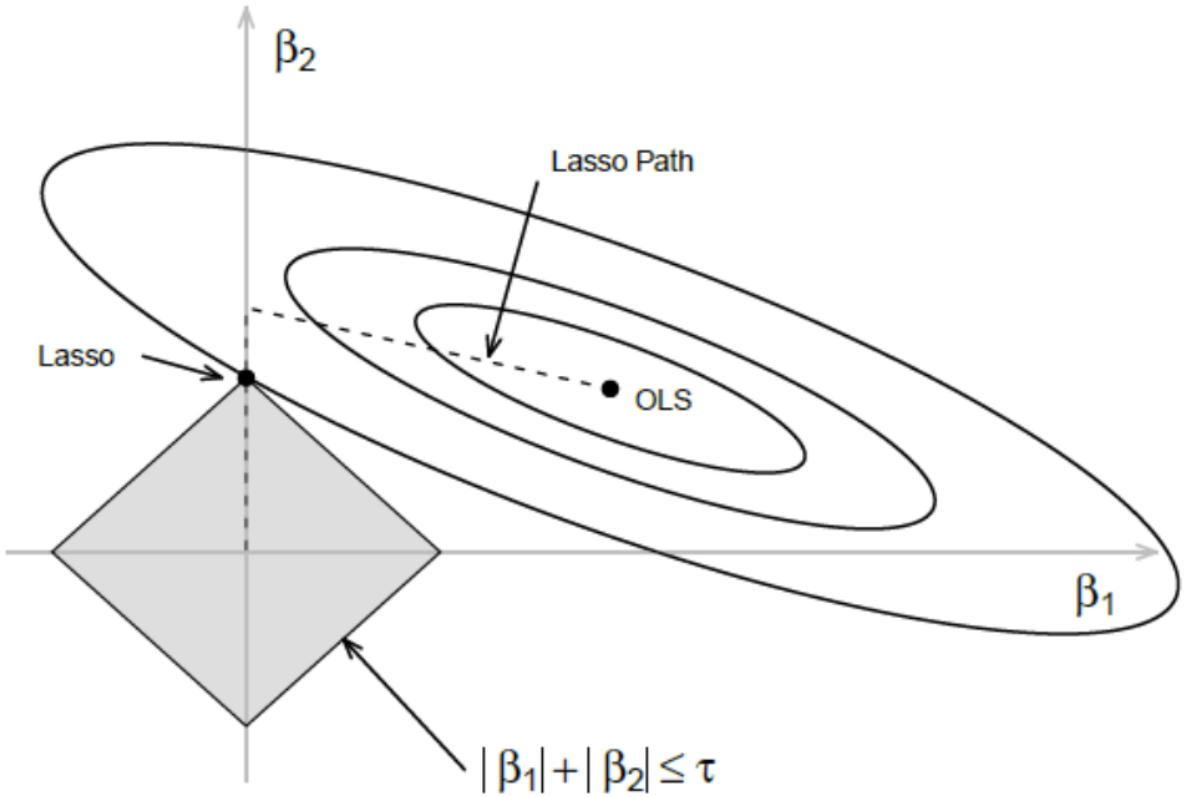


Figure 29.3: Lasso Dual Minimization Solution

The Lasso path is drawn with the dashed line. This is the sequence of solutions obtained as the constraint set is varied.

Lasso: Dual Minimization

- Since we minimize a quadratic subject to a polytope, the solution tends to be at vertices.
- This eliminates a subset of coefficients (in this case $\beta_1 = 0$).
- While Lasso is a shrinkage estimator, it does not shrink individual coefficients monotonically (in this example: as β_1 shrinks to zero, β_2 grows).
- In Ridge regression all coefficients move toward zero by a similar amount.
- In Lasso, sufficiently small coefficients are shrunken all the way to zero.

Lasso Regression

Note:

- The penalty has a different meaning **depending on the scale of the regressors**.
- Consequently, it is important to **scale the regressors** appropriately before applying Lasso.

Lasso Regression

Note:

- The penalty has a different meaning **depending on the scale of the regressors**.
- Consequently, it is important to **scale the regressors** appropriately before applying Lasso.
- It is conventional to scale all the variables to have mean zero and unit variance.

Lasso Penalty Selection

- Picking λ induces a trade-off between complexity and parsimony.
- As λ increases, the number of selected variables falls.
- K-fold selects models with good forecast accuracy, but not necessarily for other purposes such as accurate inference.
- Another popular choice is the "1se" rule: the λ which yields the most parsimonious model within one standard error of β_{ols}.
- The idea is to select a model similar to, but more parsimonious than, the CV-minimizing choice.

Code

Elastic Net

Taking a weighted average of the Ridge and Lasso penalties, we obtain the **Elastic Net** criterion:

Elastic Net

Taking a weighted average of the Ridge and Lasso penalties, we obtain the **Elastic Net** criterion:

$$SSE(\beta, \lambda, \alpha) = (Y - X\beta)'(Y - X\beta) + \lambda \left(\alpha \|\beta\|_2^2 + (1 - \alpha)\|\beta\|_1\right)$$

- With weight $0 \leq \alpha \leq 1$. Lasso ($\alpha = 0$) and Ridge ($\alpha = 1$).
- Parameters (α, λ) are selected by

Elastic Net

Taking a weighted average of the Ridge and Lasso penalties, we obtain the **Elastic Net** criterion:

$$SSE(\beta, \lambda, \alpha) = (Y - X\beta)'(Y - X\beta) + \lambda \left(\alpha \|\beta\|_2^2 + (1 - \alpha)\|\beta\|_1\right)$$

- With weight $0 \leq \alpha \leq 1$. Lasso ($\alpha = 0$) and Ridge ($\alpha = 1$).
- Parameters (α, λ) are selected by **joint minimization of the K-fold cross-validation criterion**.

Application: Lasso for Causal Inference

Belloni, Chernozhukov & Hansen (2014, *REStud*)

- We often care about one coefficient α:

$$Y_i = \alpha D_i + X_i' \beta + e_i$$

- But there are many possible controls X_i (p large).

Application: Lasso for Causal Inference

Belloni, Chernozhukov & Hansen (2014, *REStud*)

- We often care about one coefficient α:

$$Y_i = \alpha D_i + X_i' \beta + e_i$$

- But there are many possible controls X_i (p large).
- Ridge and Lasso are built for prediction, not inference.
- **Question:** Can we use Lasso to choose controls and still estimate α correctly?

Problem with Single Selection

- Naive approach: run Lasso of Y_i on X_i and keep the selected controls.

Problem with Single Selection

- Naive approach: run Lasso of Y_i on X_i and keep the selected controls.
- Lasso drops controls that predict Y only weakly.
- If such a control is strongly related to D_i, dropping it causes **omitted variable bias**.

Problem with Single Selection

- Naive approach: run Lasso of Y_i on X_i and keep the selected controls.
- Lasso drops controls that predict Y only weakly.
- If such a control is strongly related to D_i, dropping it causes **omitted variable bias**.
- **Solution:** select controls using *both* Y and D.

Post-Double-Selection

1. Lasso of Y_i on $X_i \Rightarrow$ selected controls $\hat{I}_1$.
2. Lasso of D_i on $X_i \Rightarrow$ selected controls $\hat{I}_2$.
3. OLS of Y_i on D_i and the controls in $\hat{I}_1 \cup \hat{I}_2$.

Post-Double-Selection

1. Lasso of Y_i on $X_i \Rightarrow$ selected controls $\hat{I}_1$.
2. Lasso of D_i on $X_i \Rightarrow$ selected controls $\hat{I}_2$.
3. OLS of Y_i on D_i and the controls in $\hat{I}_1 \cup \hat{I}_2$.

- Standard errors and confidence intervals for $\hat{\alpha}$ are valid.
- The choice of controls is transparent and rule-based.

Application: Abortion and Crime

- The authors revisit Donohue & Levitt (2001): did legalized abortion reduce crime in the U.S.?
- They add hundreds of possible state-level controls (initial conditions, trends, interactions).

Application: Abortion and Crime

- The authors revisit Donohue & Levitt (2001): did legalized abortion reduce crime in the U.S.?
- They add hundreds of possible state-level controls (initial conditions, trends, interactions).
- **Result:** the estimates have a similar sign, but the standard errors grow a lot, and the effects are no longer significant.
- The original result depends heavily on the choice of controls.

Takeaways

- Do not interpret Lasso or Ridge coefficients causally.
- Use Lasso to **select controls**, then estimate with OLS.
- Always select on both the outcome and the treatment.
- In Stata: `dsregress y d, controls(x1-xp)` runs double selection, where `y` is the outcome, `d` is the treatment, and `x1-xp` are the candidate controls.