Big Data in Finance

Week 2: Linear Regression Methods

Dr Daniele Bianchi

Spring 2026

Today's Agenda

OLS: Review and Limitations

Ridge Regression

Lasso Regression

Elastic Net

Selecting λ in Penalized Regressions

Application 1: Market Timing

Application 2: Factor Investing

Summary and Next Steps

Recap: The Prediction Problem

Last week we established the framework:

$$Y = f(X) + \epsilon, \qquad \mathbb{E}[\epsilon] = 0$$

Key insights from Lecture 1:

- We estimate $\widehat{f}$ from training data
- Prediction error = Bias^2 + Variance + Irreducible error
- Model complexity creates a **trade-off**
- Model must be evaluated **out-of-sample**

This week: What happens when we have *many* predictors?

The Challenge in Finance

Typical setting:

- Many candidate predictors (firm characteristics, macro variables)
- Limited time series (decades, not centuries)
- Low signal-to-noise ratio in returns (noisy returns, weak correlations)

Examples:

- Market timing: 15–20 macro predictors, 50–100 years monthly data
- Factor investing: 100+ firm characteristics, 30–60 years of data

Central Question

How do we estimate reliably when the number of (possibly weak) predictors p is large relative to sample size n?

OLS: Review and Limitations

OLS — Review and Limitations

Linear regression model:

$$y_i = x_i'\beta + \epsilon_i, \qquad i = 1, \ldots, n$$

Notation:

- y: $n \times 1$ vector of outcomes (e.g., returns)
- X: $n \times p$ matrix of predictors (e.g., characteristics, macro variables)
- β: $p \times 1$ vector of coefficients (unknown)
- ϵ: $n \times 1$ vector of errors

In matrix form:

$$y = X\beta + \epsilon$$

OLS — Estimator

Objective: Minimize sum of squared residuals (RSS)

$$\widehat{\beta}^{OLS} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i'\beta)^2 = \arg\min_{\beta} \|y - X\beta\|^2$$

Closed-form solution:

$$\widehat{\beta}^{OLS} = (X'X)^{-1}X'y$$

Requires $X'X$ to be invertible (full rank).

OLS — Properties (Review)

Under standard assumptions ($\mathbb{E}[\epsilon|X] = 0$, constant variance of ϵ):

1. **Unbiased:**
$$\mathbb{E}[\widehat{\beta}^{OLS}] = \beta$$

2. **Variance:**
$$\text{Var}(\widehat{\beta}^{OLS}) = \sigma^2(X'X)^{-1}$$

3. **Gauss-Markov:** Best Linear Unbiased Estimator (BLUE)

Key Point

OLS is optimal when its assumptions hold and p is small relative to n.

When OLS Fails — Too Many Predictors

The problem: More predictors than observations

Example:

- 100 firm characteristics (predictors), 60 months of data (observations)
- OLS cannot find a unique solution

Why? With $p > n$, the model can fit the data in countless ways.

- Mathematically: $(X'X)$ becomes non-invertible when $p > n$
- Intuitively: Too many "knobs to turn," not enough data to pin them down

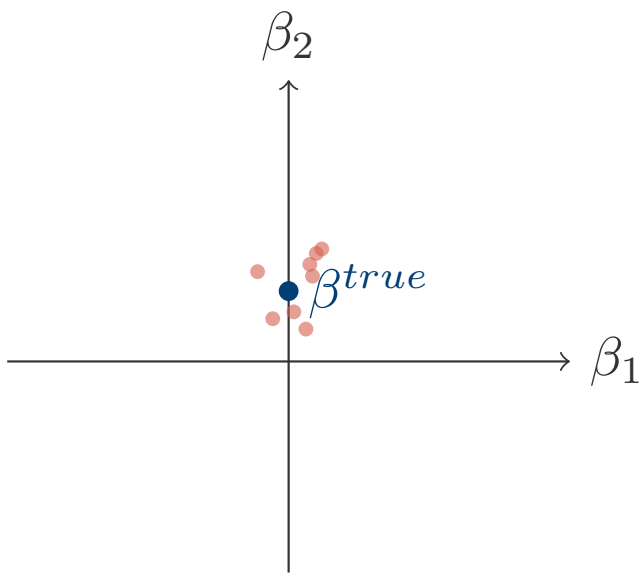
Even when $p < n$: Correlated predictors cause instability

- Small changes in data $\Rightarrow$ large swings in $\widehat{\beta}$
- Estimates are unreliable

Result: High variance, poor out-of-sample performance

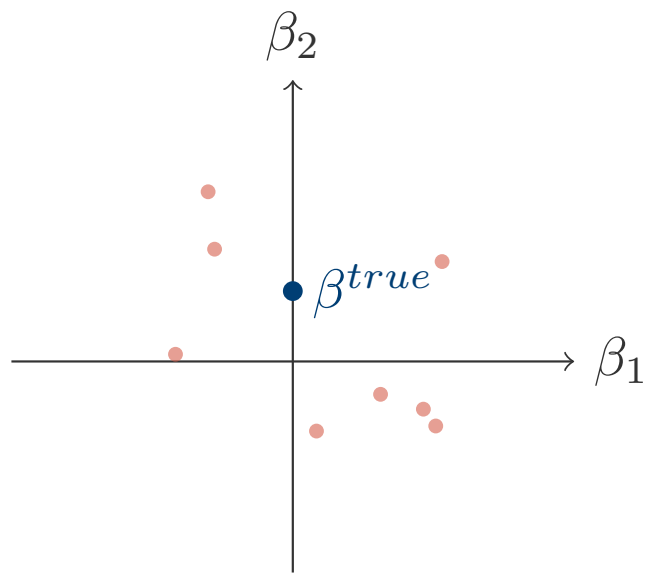
Variance Inflation — Intuition

Few predictors ($p \ll n$)



Low variance

Many predictors ($p \approx n$)



High variance

Each orange point: $\widehat{\beta}^{OLS}$ from a different sample.
With many predictors, estimates are **scattered** around the true β^{true}

The Overfitting Problem

In-sample: Adding predictors always improves fit

- R^2 increases (or stays the same)
- RSS decreases (or stays the same)
- Can achieve a perfect fit if $p = n$

Out-of-sample: Performance often *decreases*

- Model fits noise, not signal
- Predictions on new data are poor
- May be worse than simple benchmarks

Reminder: The Golden Rule

Always evaluate predictive performance **out-of-sample**.

The Solution — Constrained Estimation

Key insight: Accept some **bias** to reduce **variance**

Recall from Lecture 1:

$$\text{MSE}(\widehat{\beta}) = \text{Bias}(\widehat{\beta})^2 + \text{Var}(\widehat{\beta})$$

OLS:

- Zero bias (unbiased)
- High variance when p is large

Regularized methods:

- Shrink coefficients toward zero
- Introduces bias, but substantially reduces variance
- Net effect: Lower total MSE

Intuition: By being "skeptical" of extreme coefficient values, we avoid fitting noise in the training data.

Ridge Regression

Ridge Regression — The Idea

Problem with OLS: Coefficients can be arbitrarily large

Solution: Penalize large coefficients

- Add a "cost" for coefficient magnitude
- Model must balance fit vs. coefficient size
- Large coefficients only if strongly supported by data

Intuition:

- Without penalty: "Use any coefficient values you want"
- With penalty: "Large coefficients are expensive"

Ridge Regression — Formulation

Penalized least squares:

$$\widehat{\beta}^{Ridge} = \arg\min_{\beta} \left\{ \underbrace{\sum_{i=1}^{n}(y_i - x_i'\beta)^2}_{\text{RSS}} + \underbrace{\lambda \sum_{j=1}^{p} \beta_j^2}_{\text{Ridge Penalty}} \right\}$$

The penalty term:

- $\sum_{j=1}^{p} \beta_j^2 = \|\beta\|_2^2$ is the squared L_2 norm
- Penalizes the sum of squared coefficients
- $\lambda \geq 0$ is a **penalty parameter** that needs to be tuned

The Role of λ

The tuning parameter λ controls the **strength of regularization**:

- $\lambda = 0$: No penalty $\Rightarrow$ recover OLS
- $\lambda \to \infty$: Infinite penalty $\Rightarrow$ all $\widehat{\beta}_j \to 0$
- Intermediate λ: Balance between fit and complexity

Key Point

λ is a **hyperparameter** — not estimated from the objective function. We choose it separately (e.g., via cross-validation).

Equivalent Constrained Form

Ridge regression can be written as a **constrained optimization**:

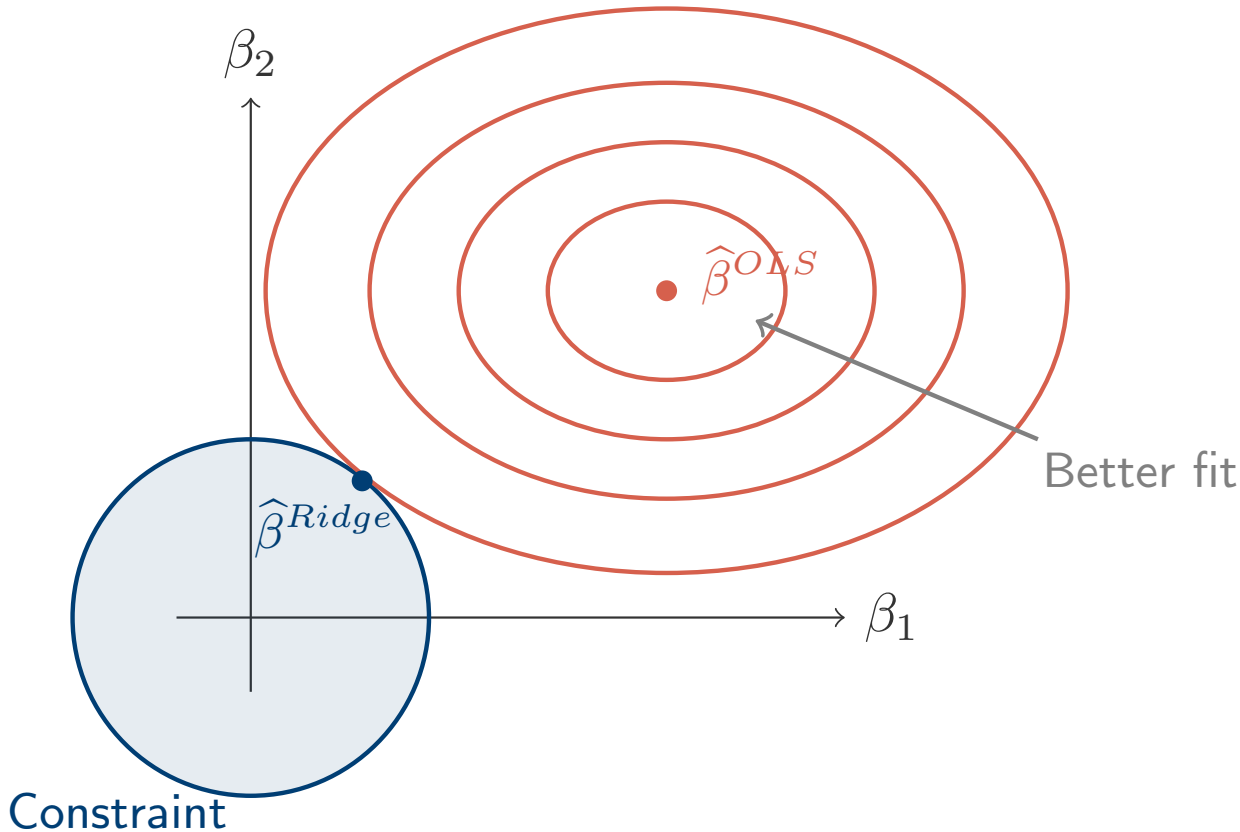
$$\min_{\beta} \sum_{i=1}^{n} (y_i - x_i'\beta)^2 \qquad \text{subject to} \qquad \sum_{j=1}^{p} \beta_j^2 \leq t$$

Interpretation:

- Minimize RSS subject to a "budget" on coefficients
- Budget t corresponds to penalty λ (one-to-one mapping)
- Smaller $t \Leftrightarrow$ larger $\lambda \Leftrightarrow$ more shrinkage

Geometric view: Coefficients must lie within a sphere of radius $\sqrt{t}$.

Geometric Interpretation



- **Ellipses:** Points with equal RSS (inner = better fit)
- **Circle:** Constraint region (coefficients must stay inside)
- **OLS:** $\widehat{\beta}^{OLS}$ best fit, ignoring the constraint
- **Ridge:** $\widehat{\beta}^{Ridge}$ best fit *within* the constraint

Closed-Form Solution

Ridge regression has an **analytical solution**:

$$\widehat{\beta}^{Ridge} = (X'X + \lambda I)^{-1} X'y$$

Compare to OLS: $\widehat{\beta}^{OLS} = (X'X)^{-1} X'y$

Key insight: Adding λI to $X'X$:

- Ensures the matrix is always invertible (even if $p > n$)
- "Stabilizes" the estimation
- Larger $\lambda \Rightarrow$ more shrinkage toward zero

Practical advantage: Closed-form solution means fast computation, even with many predictors.

What Does Ridge Do? — Shrinkage

Key properties:

- All coefficients are shrunk toward zero
- No coefficient is set exactly to zero (all predictors remain)
- Correlated predictors: Weight is shared more evenly across them

Intuition: Ridge is "skeptical" of large coefficients — requires strong evidence in the data to maintain them.

Financial interpretation:

- Large coefficient = strong belief in predictability
- Shrinkage = disciplined skepticism
- Appropriate given concerns about data mining in finance

Bias-Variance Tradeoff for Ridge

Bias: Ridge introduces bias

- $\mathbb{E}[\widehat{\beta}^{Ridge}] \neq \beta$ (biased estimator)
- Coefficients are systematically shrunk toward zero

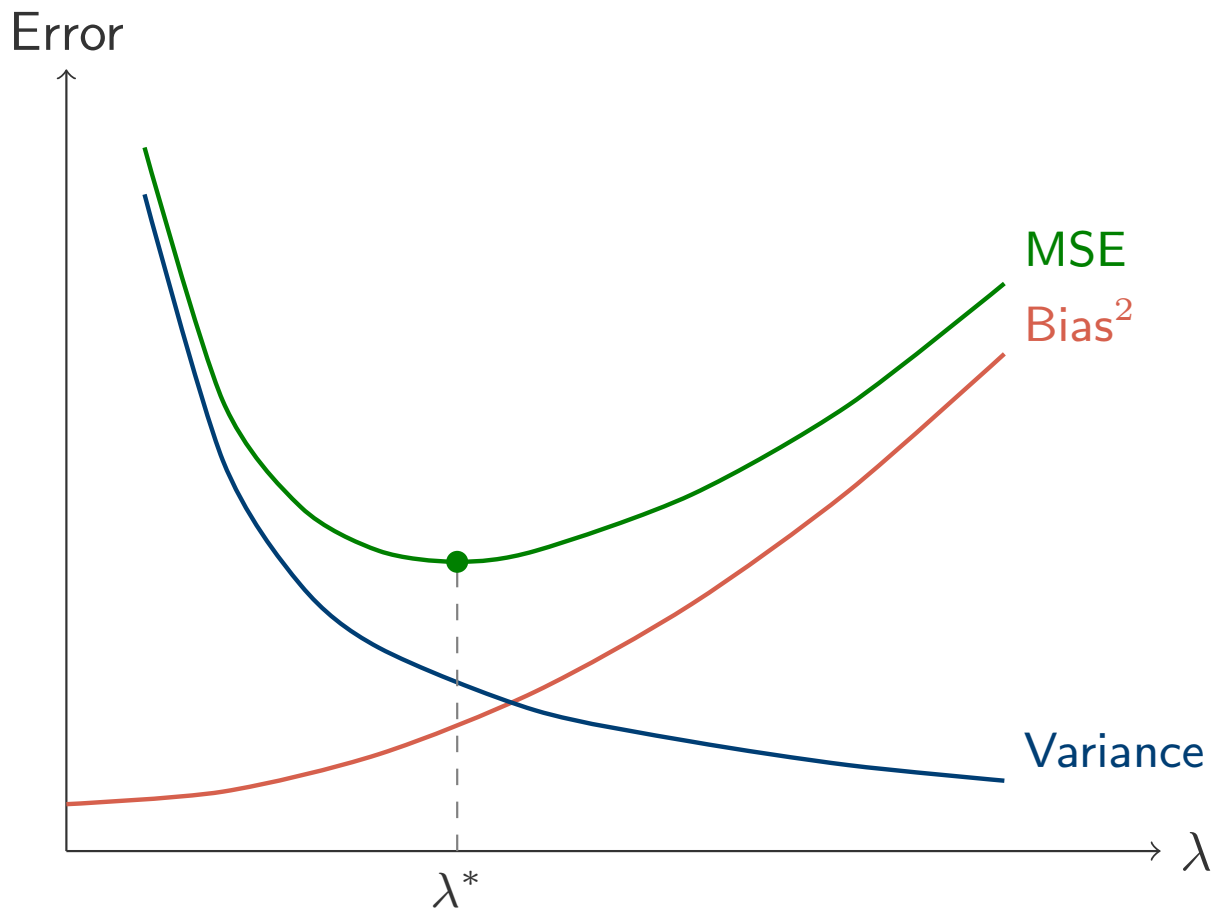
Variance: Ridge reduces variance

- $\text{Var}(\widehat{\beta}^{Ridge}) < \text{Var}(\widehat{\beta}^{OLS})$
- Shrinkage stabilizes estimates across different samples

MSE tradeoff:

- Small λ: Low bias, high variance (close to OLS)
- Large λ: High bias, low variance (close to zero)
- Optimal λ: Minimizes total MSE

MSE as a Function of λ



Optimal λ^* balances bias and variance to minimize total MSE.

Important: Feature Scaling

Problem: Ridge penalty depends on coefficient magnitude

$$\text{Penalty} = \lambda \sum_{j=1}^{p} \beta_j^2$$

If predictors are on different scales (e.g., dollars vs. millions):

- Coefficients β_j have different magnitudes
- Penalty affects them unequally

Solution: **Standardize** each feature before estimation

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}$$

- Each feature has mean zero, standard deviation one
- Coefficients become comparable
- In sklearn: `StandardScaler()` handles this automatically

Lasso Regression

Limitation of Ridge

Ridge regression:

- Shrinks all coefficients toward zero
- But: Never sets coefficients *exactly* to zero
- All p predictors remain in the model

When is this a problem?

- Many irrelevant predictors (want to exclude them)
- Desire for interpretability (which variables matter?)
- Sparse models may generalize better out-of-sample

Want: A method that performs **variable selection** automatically.

Lasso — Formulation

Least Absolute Shrinkage and Selection Operator (Tibshirani, 1996):

$$\widehat{\beta}^{Lasso} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

Key difference from Ridge:

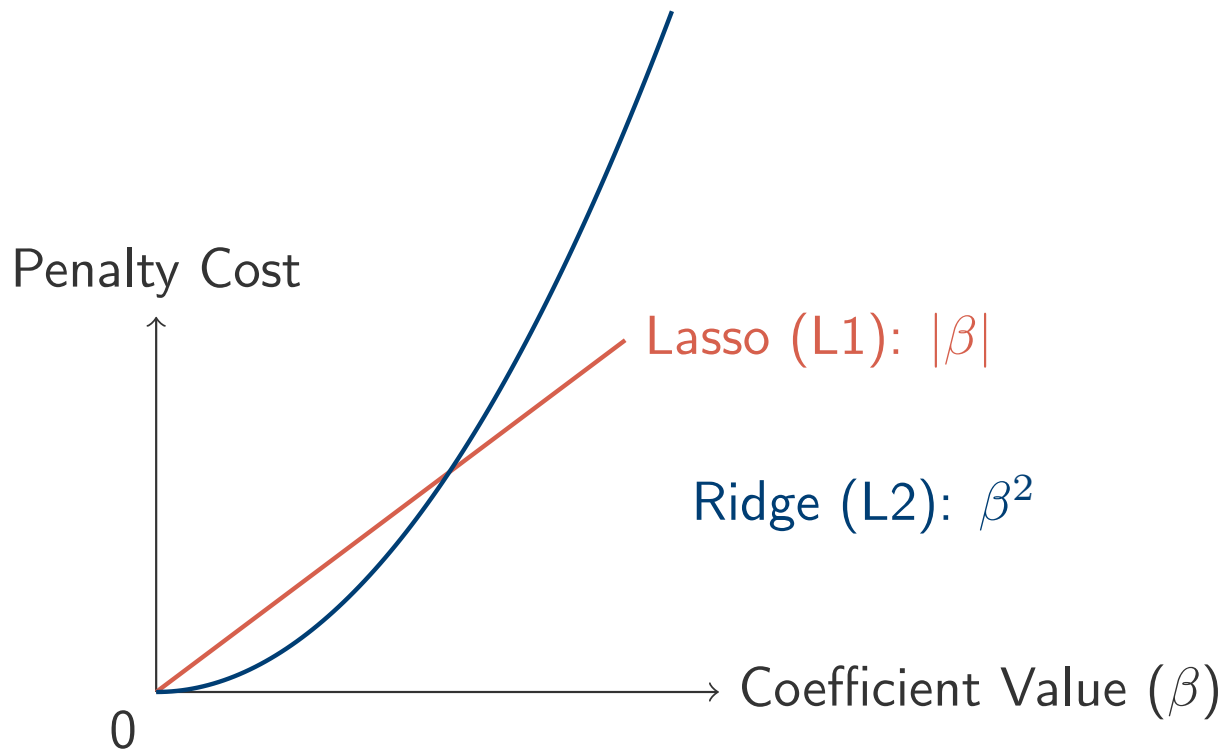
- Ridge: L_2 penalty $\sum_j \beta_j^2$ (squared coefficients)
- Lasso: L_1 penalty $\sum_j |\beta_j|$ (absolute values)

Crucial property: Lasso produces **sparse** solutions — some $\widehat{\beta}_j = 0$ exactly.

The Key Difference: Ridge vs. Lasso Penalties

Ridge Regression (L2)	**Lasso Regression (L1)**
Penalizes **squares**: β^2	Penalizes **absolute values**: $\lvert\beta\rvert$
Shrinks coefficients proportionally	Pushes coefficients toward zero
Never becomes exactly zero	**Can become exactly zero**

Visualizing the Penalty Effect:



How Lasso Works: The "Soft Thresholding" Rule

A Simple Decision Rule:
Lasso asks each variable: *"Is your signal strong enough to survive the penalty?"*

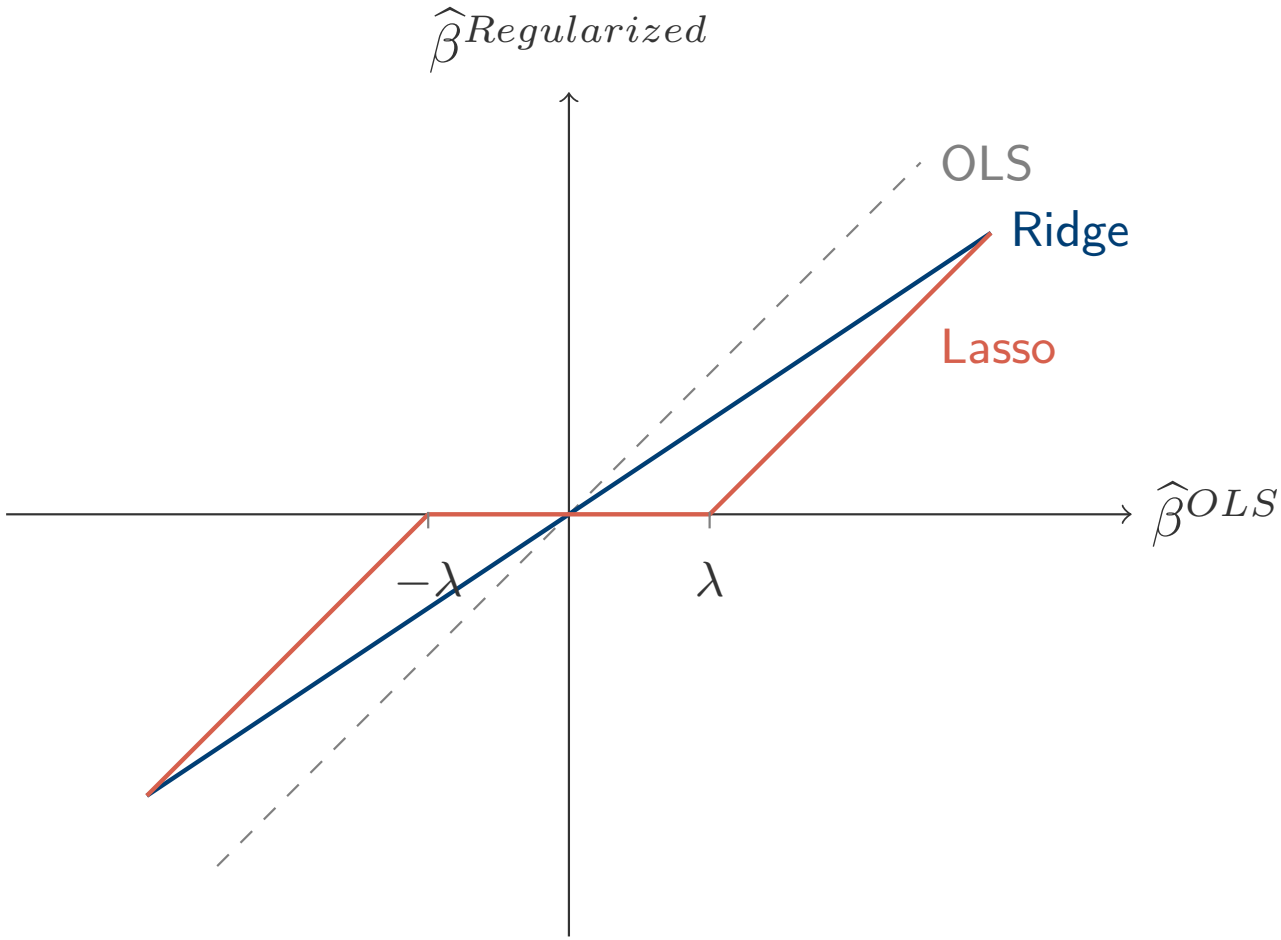
For each coefficient:

1. Start with the regular model coefficient ($\widehat{\beta}_{OLS}$)
2. Subtract the Lasso penalty (λ)
3. Apply the decision rule:
   - If result is **positive** $\rightarrow$ keep it
   - If result is **negative** $\rightarrow$ set to **zero**

Example: (with penalty $\lambda = 0.5$)

	Step 1: $\widehat{\beta}_{OLS}$	**Step 2:** Subtract λ	**Step 3:** Final Decision
Variable A	1.2	1.2 - 0.5 = 0.7	**Kept** (0.7)
Variable B	0.4	0.4 - 0.5 = -0.1	**Set to ZERO**
Variable C	-0.8	|-0.8| - 0.5 = 0.3	**Kept** (-0.3)

Shrinkage Comparison: Ridge vs. Lasso



Ridge: Proportional shrinkage (all coefficients reduced).
Lasso: Soft thresholding (small coefficients set to zero).

Lasso — No Closed-Form Solution

General case: No analytical solution for Lasso

Why?

- The L_1 penalty $|\beta_j|$ is not differentiable at zero
- Cannot set derivative to zero and solve

Solution: Iterative optimization algorithms

- Coordinate descent: Update one β_j at a time
- Very efficient implementations exist
- sklearn uses fast coordinate descent

Practical implementation: Slightly slower than Ridge, but still fast.

Lasso — Properties

Advantages:

- Automatic variable selection (sparse solutions)
- Interpretability: Identifies "important" predictors
- Can handle $p > n$ (selects at most n variables)

Limitations:

- With correlated predictors: Selects one arbitrarily
- Can be unstable: Different samples $\Rightarrow$ different selections
- At most $\min(n, p)$ non-zero coefficients

Next: what to do when correlation matters but we still want to do variable selection.

Elastic Net

Combining Ridge and Lasso

Ridge:

+ Handles correlated predictors well
+ Stable estimates
– No variable selection (keeps all predictors)

Lasso:

+ Automatic variable selection
+ Sparse, interpretable models
– Unstable with correlated predictors

Elastic Net: Combine both penalties to get the best of both worlds.

Elastic Net — Formulation

Zou & Hastie (2005):

$$\widehat{\beta}^{EN} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \left[\alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2 \right] \right\}$$

Two hyperparameters:

- $\lambda \geq 0$: Overall penalty strength
- $\alpha \in [0, 1]$: Mixing parameter
  - $\alpha = 1$: Pure Lasso
  - $\alpha = 0$: Pure Ridge
  - $\alpha \in (0, 1)$: Combination

Elastic Net — Properties

Inherits advantages of both methods:

From Lasso (L_1 component):

- Sparse solutions (variable selection)
- Can set coefficients exactly to zero

From Ridge (L_2 component):

- **Grouping effect**: Correlated predictors selected together
- More stable than pure Lasso
- Can select more than n variables

In practice: Often the best choice for high-dimensional problems.

Method Comparison

Method	Penalty	Sparse?	Closed-form?	Correlation
OLS	None	No	Yes	Poor
Ridge	L_2	No	Yes	Good
Lasso	L_1	Yes	No	Poor
Elastic Net	$L_1 + L_2$	Yes	No	Good

Guidance:

- Few predictors, low correlation $\Rightarrow$ OLS may suffice
- Many correlated predictors $\Rightarrow$ Ridge or Elastic Net
- Want variable selection $\Rightarrow$ Lasso or Elastic Net
- High-dimensional, correlated $\Rightarrow$ Elastic Net

Recap: Key Equations

OLS:

$$\widehat{\beta}^{OLS} = (X'X)^{-1}X'y$$

Ridge:

$$\widehat{\beta}^{Ridge} = (X'X + \lambda I)^{-1}X'y$$

Lasso:

$$\widehat{\beta}^{Lasso} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n}(y_i - x_i'\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

Elastic Net:

$$\widehat{\beta}^{EN} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n}(y_i - x_i'\beta)^2 + \lambda \left[\alpha \sum_{j=1}^{p} |\beta_j| + (1-\alpha) \sum_{j=1}^{p} \beta_j^2 \right] \right\}$$

Selecting λ in Penalized Regressions

The Critical Question

Remember: Ridge, Lasso, Elastic Net all depend on λ (and α).

How do we choose these hyperparameters?

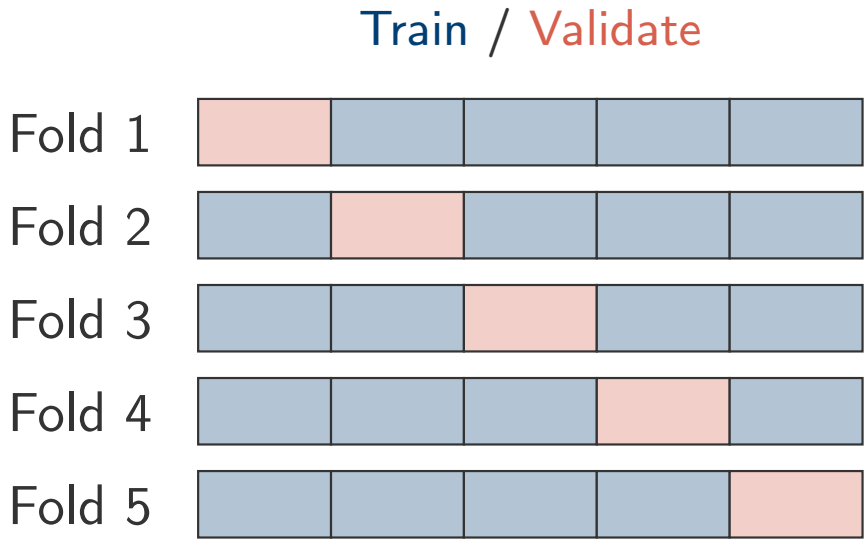
Cannot use training sample:

- Training error always decreases with less regularization
- Would select $\lambda = 0$ (back to OLS)
- Overfits to training data

Need: A proxy for the **out-of-sample** performance of the model.

Cross-Validation: Review

K-**Fold Cross-Validation**



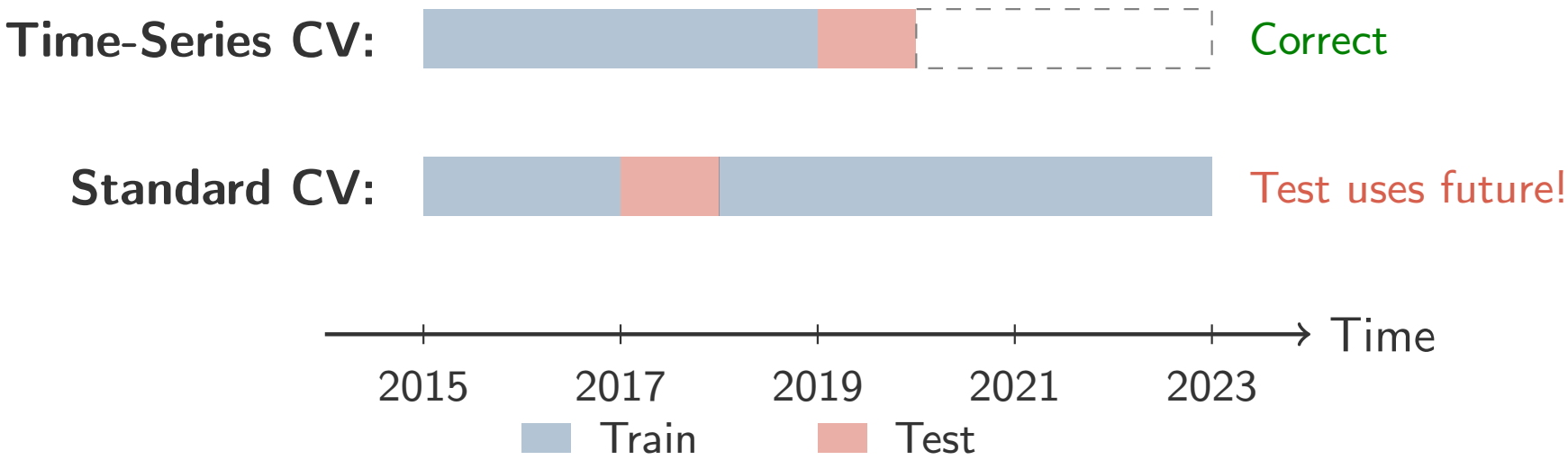
Procedure:

1. Split data into K equal parts ("folds")
2. For each fold: train on other K-1 folds, validate the prediction error on this fold
3. Average the K prediction errors

Common choice: $K = 5$ or $K = 10$

Problem: Time Series Structure

Standard CV: Observations are exchangeable (order does not matter).



Time series reality:

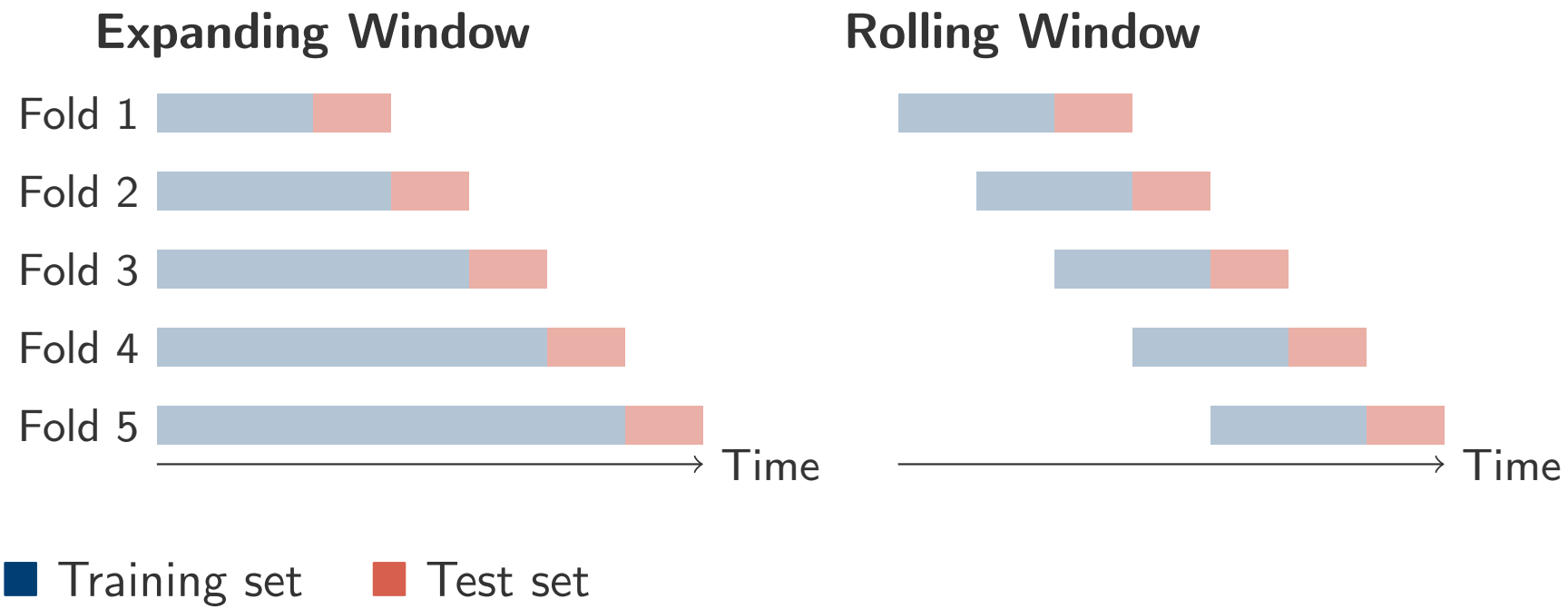
- Observations are ordered in time
- Cannot use future data to predict past

Random splits create look-ahead bias:

- Model trained on 2015–2020 data
- Tested on 2018 data (performance is biased!)

Time-Series Cross-Validation

Key principle: Always train on past, test on future.



Both respect temporal ordering — no look-ahead bias.

- **Expanding window:** More data as we move forward; assumes stable relationships
- **Rolling window:** Older data dropped as new data added; adapts to regime changes

Choosing λ: Procedure

Step 1: Define a grid of candidate λ values

- Spaced on log scale (factors of 10): $\lambda \in \{0.001, 0.01, 0.1, 1, 10, 100\}$

Step 2: For each λ, compute time-series CV error

- Use 5-fold time-series CV on training data
- For each fold k: calculate prediction error $\text{MSE}_k(\lambda)$
- Average across folds: $\overline{\text{CV}}(\lambda) = \frac{1}{5}\sum_{k=1}^{5} \text{MSE}_k(\lambda)$
- **Standard error**: $\text{SE}(\lambda) = \sqrt{\frac{1}{5}\sum_{k=1}^{5}(\text{MSE}_k(\lambda) - \overline{\text{CV}}(\lambda))^2}$

Step 3: Plot CV error vs. λ with error bars

Step 4: Select optimal λ

- Option A: $\lambda_{\min}$ that minimizes $\overline{\text{CV}}(\lambda)$
- Option B: Largest λ where $\overline{\text{CV}}(\lambda) \leq \overline{\text{CV}}(\lambda_{\min}) + \text{SE}(\lambda_{\min})$

Understanding the Standard Error in CV

Why do we need the standard error?

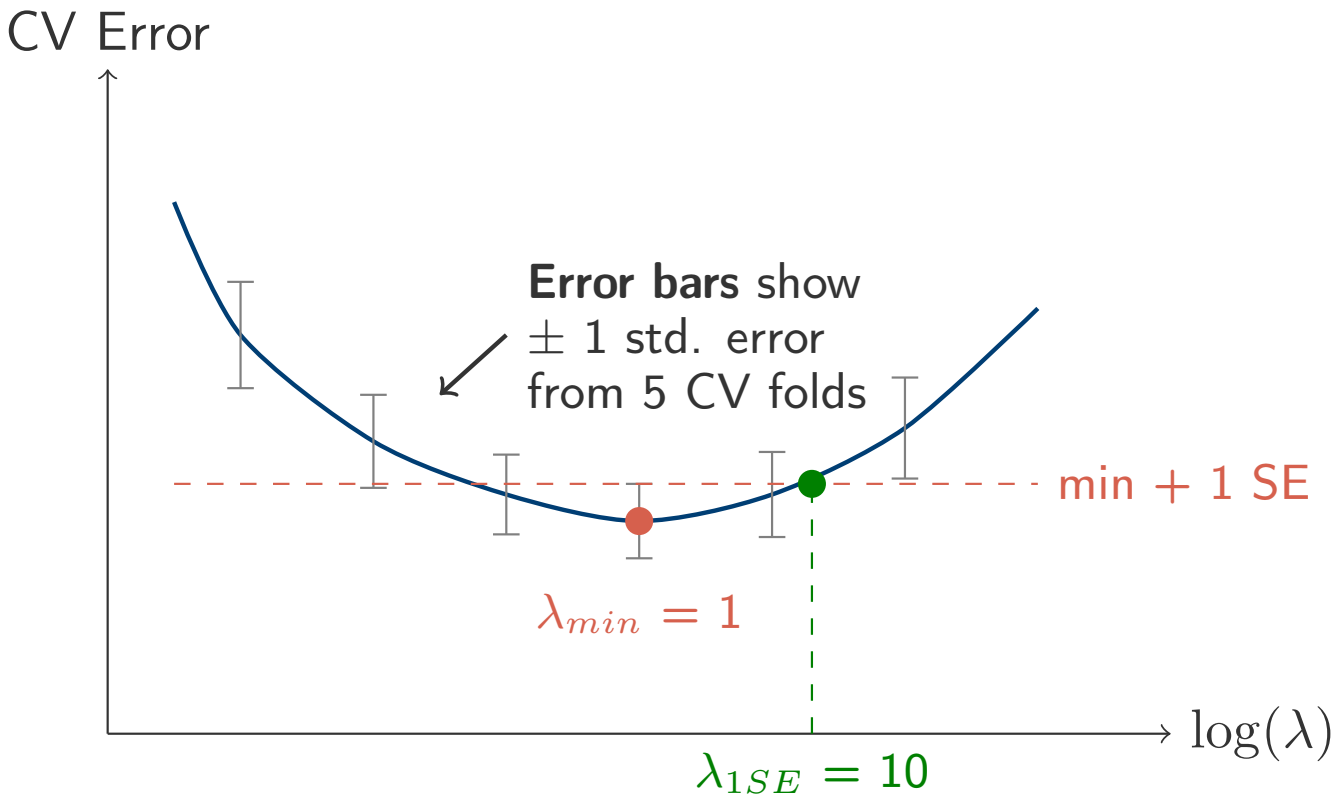
- CV error is computed on **5 different validation folds**
- Each fold gives a slightly different error estimate
- The **standard error measures variability** across folds

Example for Ridge with $\lambda = 10$:

Fold	MSE
Fold 1	0.0042
Fold 2	0.0038
Fold 3	0.0045
Fold 4	0.0041
Fold 5	0.0039
Mean	0.0041
Std. Dev.	0.00026
Std. Error (SE)	**0.00012**

Note: SE = Std. Dev. / $\sqrt{5}$ measures uncertainty in mean CV error

Choosing λ: The 1-Standard-Error Rule



1-SE rule intuition: If $\lambda = 10$ performs within 1 SE of $\lambda = 1$, the difference could be **just noise from CV sampling**. Choose the simpler model ($\lambda = 10 =$ more regularization).

Why Use the 1-SE Rule?

Motivation: Protect against overfitting to the validation set

$\lambda_{\min}$ **(minimum CV error):**

+ Best on validation data
– May overfit to noise in validation folds
– Less regularization
– More complex model

λ_{1SE} **(1-SE rule):**

+ More conservative
+ Simpler model (more shrinkage)
+ Often better on true test data
– Slightly worse on validation

Practical Advice

- **Default:** Use λ_{1SE} for most applications
- **Alternative:** Use $\lambda_{\min}$ if you have lots of data and computational budget for extensive CV
- **Both:** Try both and compare on final test set

Example: Ridge Regression CV Results

Suppose we get these results from 5-fold time-series CV:

λ	Mean CV Error	Std Error	Min + 1 SE
0.001	0.0052	0.00018	—
0.01	0.0048	0.00016	—
0.1	0.0044	0.00015	—
1	**0.0041**	0.00012	0.0041 + 0.00012 = **0.00422**
10	0.0042	0.00013	—
100	0.0046	0.00014	—

- $\lambda_{\min}$**:** $\lambda = 1$ has lowest CV error (0.0041)
- **1-SE threshold:** 0.0041 + 0.00012 = 0.00422
- λ_{1SE}**:** $\lambda = 10$ has CV error 0.0042 $<$ 0.00422 ✓
- Higher λ (100) exceeds threshold, so choose $\lambda = 10$

Interpretation: $\lambda = 10$ is statistically indistinguishable from $\lambda = 1$, but simpler (more regularization), so we prefer it.

Application 1: Market Timing

The Equity Premium Prediction Problem

Central question: Can we predict the stock market?

If YES:

- Time the market (shift allocation between stocks and bonds)
- Significant economic value

If NO:

- Buy and hold is optimal
- Active timing destroys value (transaction costs)

Academic debate: Decades of research, no consensus.

The Prediction Model

Time-series regression:

$$r_{m,t+1} = \alpha + x_t' \boldsymbol{\beta} + \epsilon_{t+1}$$

Variables:

- $r_{m,t+1}$: Market excess return in month $t+1$
- x_t: Vector of predictor variables known at time t
- $\boldsymbol{\beta}$: Coefficients capturing predictive relationships

Goal: Forecast $r_{m,t+1}$ using information available at t.

Classical predictors: $\sim$15–20 variables commonly used in the literature

- Valuation ratios: Dividend-price ratio (D/P), Earnings-price ratio (E/P), Book-to-market ratio (B/M)
- Interest rates: Treasury bill rate, Term spread (long minus short rates), Default spread (BAA minus AAA yields)
- Other: Stock variance, Inflation, Net equity expansion

The Benchmark: Historical Mean

Simplest possible forecast:

$$\widehat{r}_{t+1} = \bar{r}_t = \frac{1}{t}\sum_{s=1}^{t} r_s$$

Expanding-window average of past returns (Campbell and Thompson 2008).

Economic underpinning:

- No predictability
- Constant expected return
- Ignores all predictor variables

Key Insight

This naive benchmark is **surprisingly hard to beat** out-of-sample.

Measuring Forecast Quality: R^2_{OS}

Out-of-sample R^2:

$$R^2_{OS} = 1 - \frac{\sum_{t=t_0}^{T-1}(r_{t+1} - \widehat{r}_{t+1})^2}{\sum_{t=t_0}^{T-1}(r_{t+1} - \bar{r}_t)^2}$$

Interpretation:

- $R^2_{OS} > 0$: Model beats the historical mean
- $R^2_{OS} = 0$: Model equals the historical mean
- $R^2_{OS} < 0$: Model is *worse* than the mean

Key difference from in-sample R^2:

- In-sample $R^2 \geq 0$ always
- Out-of-sample R^2_{OS} can be negative!

Goyal & Welch (2008): Original Findings

Comprehensive study of equity premium prediction.

Methodology:

- Test all major predictors from the literature
- Proper out-of-sample evaluation
- Expanding window estimation

Results:

- Most predictors: $R^2_{OS} < 0$
- In-sample significance $\neq$ out-of-sample success
- Historical mean hard to beat

Sobering conclusion: Market timing is very difficult.

Goyal, Welch & Zafirov (2024): Updated Evidence

Extended sample: Through 2022 (15 more years of data).

Key questions:

- Do original findings hold?
- Have any predictors improved?
- What about new predictors?

Main findings:

- Original conclusions largely confirmed
- Most predictors still fail out-of-sample
- A few show modest improvement
- Still very difficult to beat the mean

The Overfitting Problem in Market Timing

With many predictors:

- OLS finds patterns that fit historical data
- Many of these patterns are spurious (noise)
- Out-of-sample: Spurious patterns don't repeat

Result:

- In-sample R^2 can be substantial (5–10%)
- Out-of-sample R^2_{OS} often negative
- Would have been better to predict zero!

Key insight: Regularization provides **disciplined skepticism** on predictability.

Revisiting Equity Market Timing

Research Question: Can we predict equity market returns using macroeconomic variables?

- Classic problem in finance: market timing
- Key tension: **many predictors** vs **limited data**
- Traditional approach (OLS) typically fails out-of-sample

This Study

Compare OLS with regularized methods (Ridge, Lasso, Elastic Net) for predicting monthly excess returns on the S&P 500 using 25 macroeconomic predictors

Key Readings: Goyal, Welch & Zafirov (2024, RFS); Campbell & Thompson (2008, RFS)

Data: GWZ (2024) Dataset

Target variable:

- Monthly excess return on S&P 500; Equity Risk Premium (ERP)
- Sample: April 1953 – December 2021 (824 months)

25 Predictors:

- **Macroeconomic variables:** Debt-to-GDP (dtoy), Inflation (infl), Output gap (ogap), Term spread (tms), Treasury bill rate (tbl), Oil price (wtexas), etc.
- **Market variables:** Average correlation (avgcor), Book-to-market (b_m), Debt-to-equity (d_e), Dividend-price (d_p), Default spread (dfy), Debt-to-assets (dtoat), Earnings-price (e_p), Net equity issuance (ntis), Stock variance (svar), Skewness (skvw), Tail risk (tail), etc.

Predictive regression: All predictors are **known only up to time** $t-1$ when forecasting return at t

Out-of-Sample ERP Predictability

Walk-Forward Predictions: The gold standard for evaluating predictive models in finance.

- Avoids look-ahead bias
- Mimics real-time trading decisions
- Tests true predictive power, not in-sample fit

Empirical Setup:

1. Initial training window: 120 months (10 years)
2. Move forward: Expand training window, predict next month
3. Out-of-sample period: 1967–2023 (672 months)
4. Benchmark: Historical mean $\widehat{r}_{t+1} = \frac{1}{t}\sum_{s=1}^{t} r_s$
5. Competing models: Ridge, Lasso, Elastic Net.
6. Hyperparameters: Selected via 5-fold time-series cross-validation on training data

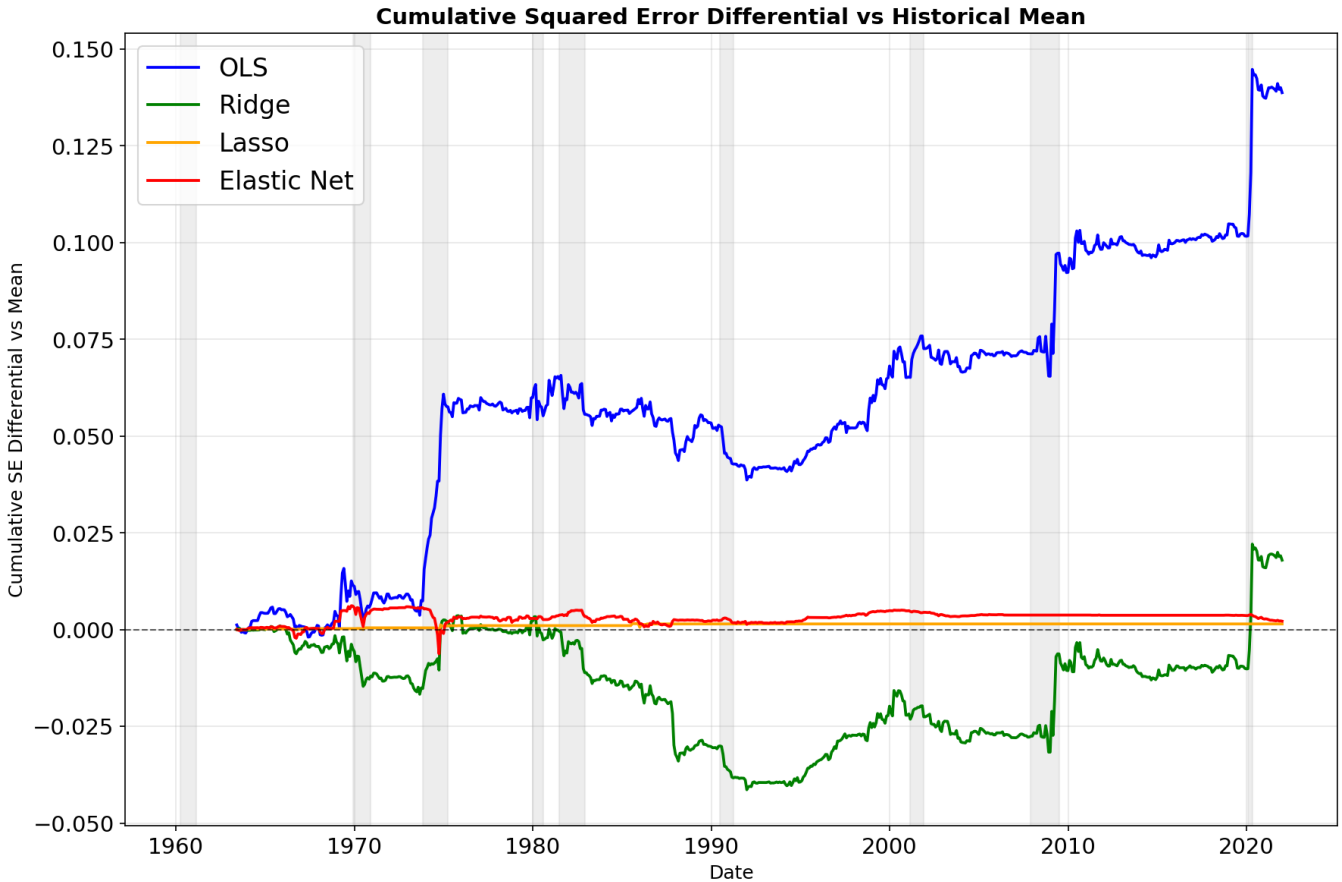
Out-of-Sample R^2 Results

Measuring predictive accuracy relative to historical mean:

Model	R^2_{OS} **(%)**
OLS	-10.69
Ridge	-1.38
Lasso	-0.11
Elastic Net	-0.17

Key insight: All models underperform historical mean, but **regularization dramatically reduces overfitting** (OLS is 100$\times$ worse than Lasso!)

Cumulative Squared Error Differential



The chart shows the **cumulative Sum of Squared Error (SSE)** difference: $\sum_{t=1}^{T}[(y_t - \widehat{y}_t^{\text{model}})^2 - (y_t - \bar{y}_t)^2]$

- **Below zero** = model beats historical mean
- **Above zero** = model worse than historical mean

Cumulative SSE Differential: What We Learn

OLS (blue):

- Starts well in 1970s, then **catastrophically diverges**
- Post-2008: Accumulates 0.10+ in squared errors vs mean
- Problem: Extreme predictions amplify errors during crises

Ridge (green):

- **Stays below zero throughout entire period**
- Occasionally beats mean (1980s-90s, 2020s)
- Most stable predictor – never catastrophically wrong

Lasso/Elastic Net (orange/red):

- Track historical mean closely (near zero line)
- Slight improvement in the 2020s
- Conservative: avoid big losses but also miss opportunities

Recessions (gray bars): All models struggle, but Ridge degrades least

Practical Lessons from Market Timing

Why are all R^2_{OS} negative?

- Market returns are **extremely difficult to predict**
- Even small forecast errors are heavily penalized
- Historical mean is a surprisingly strong benchmark
- This is **typical** in equity premium prediction literature

What matters:

1. **Relative performance:** Regularized models much closer to zero
2. **Economic value:** Can poor R^2 still generate profits?
3. **Directional accuracy:** Getting the sign right matters more than magnitude

Sign Accuracy Analysis

Getting the direction right is crucial for market timing:

Model	Sign Acc. (%)	When Up (%)	When Down (%)
Historical Mean	59.4	100.0	0.0
OLS	57.8	68.7	42.0
Ridge	**59.5**	73.9	38.5
Lasso	59.2	99.8	0.0
Elastic Net	58.8	94.3	7.0

Observations:

- Ridge has **best overall sign accuracy** (59.5%)
- Lasso/Elastic Net behaves like the historical mean (always bullish)
- OLS is more balanced but least accurate overall
- **Important:** 59.5% accuracy can be economically valuable!

From Predictions to Portfolio Weights

Market timing strategy: Allocate between equity and risk-free asset

Mean-variance optimal weight:

$$\omega_t = \frac{1}{\gamma} \frac{\widehat{r}_{t+1}}{\widehat{\sigma}_t^2}$$

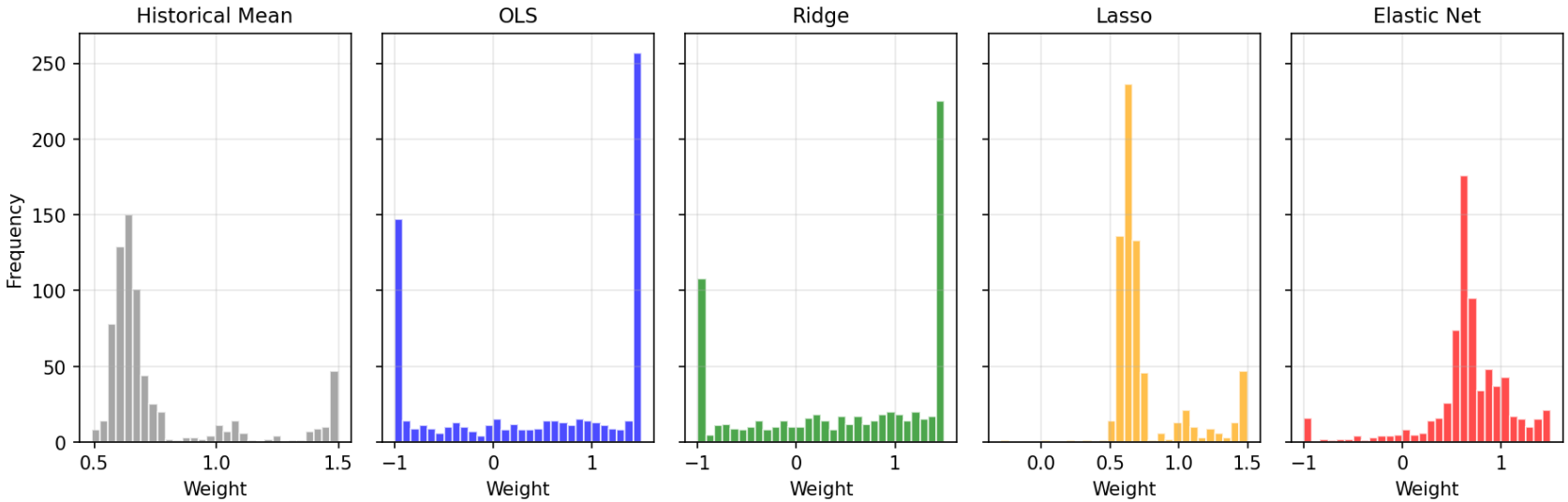
where:

- $\widehat{r}_{t+1}$ = model's predicted excess return
- $\gamma = 5$ = risk aversion coefficient
- $\widehat{\sigma}_t^2$ = realized variance (rolling 60-month window)
- Constraints: $-1 \leq \omega_t \leq 1.5$ (allow 50% leverage, 100% cash)

Portfolio return:

$$r_{t+1}^{\text{portfolio}} = \omega_t \cdot r_{t+1}^{\text{market}} + (1 - \omega_t) \cdot r_f$$

Portfolio Weights Distribution



What are we looking at? Histogram of equity weights over 672 months

- OLS and Ridge weights often at constraints ($\omega = -1$ or $\omega = 1.5$)
- Lasso weights resembles the historical mean's: strong L1 penalty often zeros out most coefficients $\Rightarrow$ forecast $\approx$ mean
- E-net is between Lasso and Ridge, with weights more dispersed around 0.75

Performance Metrics Summary

Gross performance metrics:

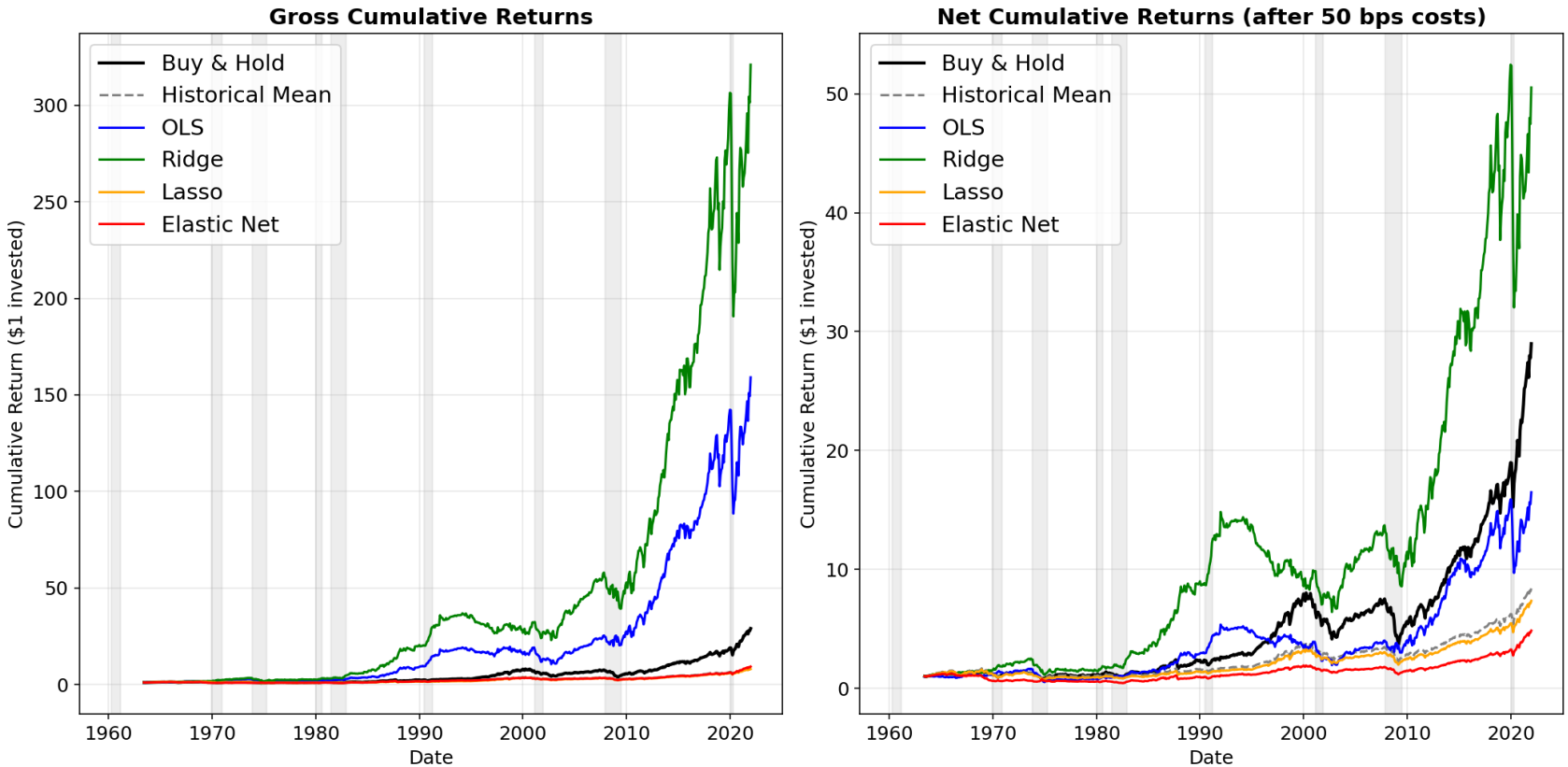
Strategy	Ann. Ret. (%)	Sharpe	Max DD (%)	Turnover	Win Rate (%)
Buy & Hold	6.86	0.46	-54.3	0.00	—
Historical Mean	4.32	0.39	-50.8	0.13	42.0
OLS	10.06	0.60	-65.6	**7.77**	54.8
Ridge	**11.14**	**0.70**	**-51.0**	6.34	**55.3**
Lasso	4.14	0.38	-50.6	0.28	42.0
Elastic Net	4.47	0.39	-44.6	2.20	42.2

After 50 bps transaction costs for each rebalancing month:

Strategy	Ann. Ret. Net (%)	Sharpe Net
Buy & Hold	6.86	0.46
OLS	6.17	0.37
Ridge	**7.97**	**0.50**

Takeaway: Ridge achieves best risk-adjusted returns even after costs

Cumulative Returns



Left panel: Gross returns (no transaction costs)
Right panel: Net returns (50 bps costs)

Net Returns: Reality Check (Right Panel)

After 50 bps (0.5%) transaction costs per trade:

OLS collapses:

- **\$158 $\rightarrow$ \$15**
- High turnover (7.8$\times$ per year) $\Rightarrow$ costs destroy value
- **Lesson:** Overconfident predictions $\Rightarrow$ excessive trading

Ridge survives:

- **\$320 $\rightarrow$ \$49**
- Moderate turnover (6.3$\times$ per year) is still costly, but tolerable
- **Still beats buy-and-hold after costs!**

Historical Mean/Lasso/Elastic Net:

- Largely unchanged (low turnover $<$ 2$\times$ per year)
- But still underperform buy-and-hold

Key Lessons for ML in Finance

1. **Regularization is essential**
   - OLS catastrophically overfits with many predictors (despite $p << n$).
   - Ridge provides the best balance: shrinkage without extreme sparsity

2. **Statistical vs economic performance diverge**
   - Negative R^2_{OS} does not mean no economic value
   - 59.5% directional accuracy $\Rightarrow$ 11% annualized return

3. **Transaction costs matter enormously**
   - High-turnover strategies (OLS) lose appeal after costs
   - More stable predictions (Ridge) better preserve value

4. **Walk-forward testing is non-negotiable**
   - In-sample R^2 would be misleading
   - Real-time prediction reveals true model performance

Practical Implementation Considerations

What we learned about deployment:

- **Feature engineering:** Use lagged predictors to avoid look-ahead bias
  - Predict r_{t+1} using X_t (known at time t)
  - Easy to get wrong – always validate timing!
- **Hyperparameter tuning:** Use time-series cross-validation
  - Never shuffle time series data
  - Refit periodically as markets evolve
- **Scaling:** Standardize features for regularized methods
  - Ridge/Lasso/Elastic Net need consistent scales
  - OLS is scale-invariant (no scaling needed)
- **Constraints:** Impose realistic portfolio constraints
  - Leverage limits, no-short-selling, etc.
  - Unconstrained weights can be extreme and unrealistic

Application 2: Factor Investing

A Different Question

Market timing: *When* to invest?

- Predict aggregate market return
- Shift between stocks and bonds

Cross-sectional prediction: *Which stocks* to pick?

- Predict relative performance across stocks
- Long winners, short losers

Key difference:

- Same methods, different data structure
- Cross-section: Much more data (many stocks)
- Often more successful than market timing

The Factor Investing Framework

Traditional approach:

1. Identify a characteristic (e.g., book-to-market)
2. Rank stocks by that characteristic
3. Long high-ranked stocks, short low-ranked
4. Earn the "factor premium"

Examples:

- Value: High book-to-market outperforms
- Momentum: Past winners continue to win
- Size: Small stocks outperform (historically)
- Profitability: Profitable firms outperform

Question: Which characteristics? How to combine them?

Data Structure: Panel

Cross-sectional regression:

$$r_{i,t+1} = x'_{i,t}\boldsymbol{\beta} + \epsilon_{i,t+1}$$

Interpretation:

- Characteristics $x_{i,t}$ predict returns at $t+1$
- β_j: Premium associated with characteristic j
- Linear combination of characteristics $\Rightarrow$ expected return

Panel structure: Observations are stock-months.

- Many more observations ($N \times T$ vs. T)
- More power to detect patterns (number of stocks N can be large)

Challenge:

- Many candidate characteristics (p large)
- Characteristics are correlated
- Which ones really matter?

The "Zoo" of Characteristics

Harvey, Liu & Zhu (2016):

- 300+ factors published in academic journals
- Most discovered through data mining
- Many are likely spurious

Gu, Kelly & Xiu (2020):

- 900+ firm characteristics
- Comprehensive dataset for ML

Problem:

- Too many predictors for OLS
- Many are correlated (size, value, profitability overlap)
- Need principled way to select/combine

Kozak, Nagel & Santosh (2020): Key Insight

Traditional approach:

- Select a small number of characteristics
- Justify selection with theory or prior evidence
- Problem: Which ones? Selection is arbitrary.

Kozak, Nagel & Santosh (KNS) approach:

- Use *all* $100+$ characteristics
- Apply Ridge regression to handle high dimensionality
- Let shrinkage determine relative importance

Key insight: Ridge shrinkage is equivalent to imposing a **prior belief** that most characteristics have small effects.

From Predictions to Portfolios

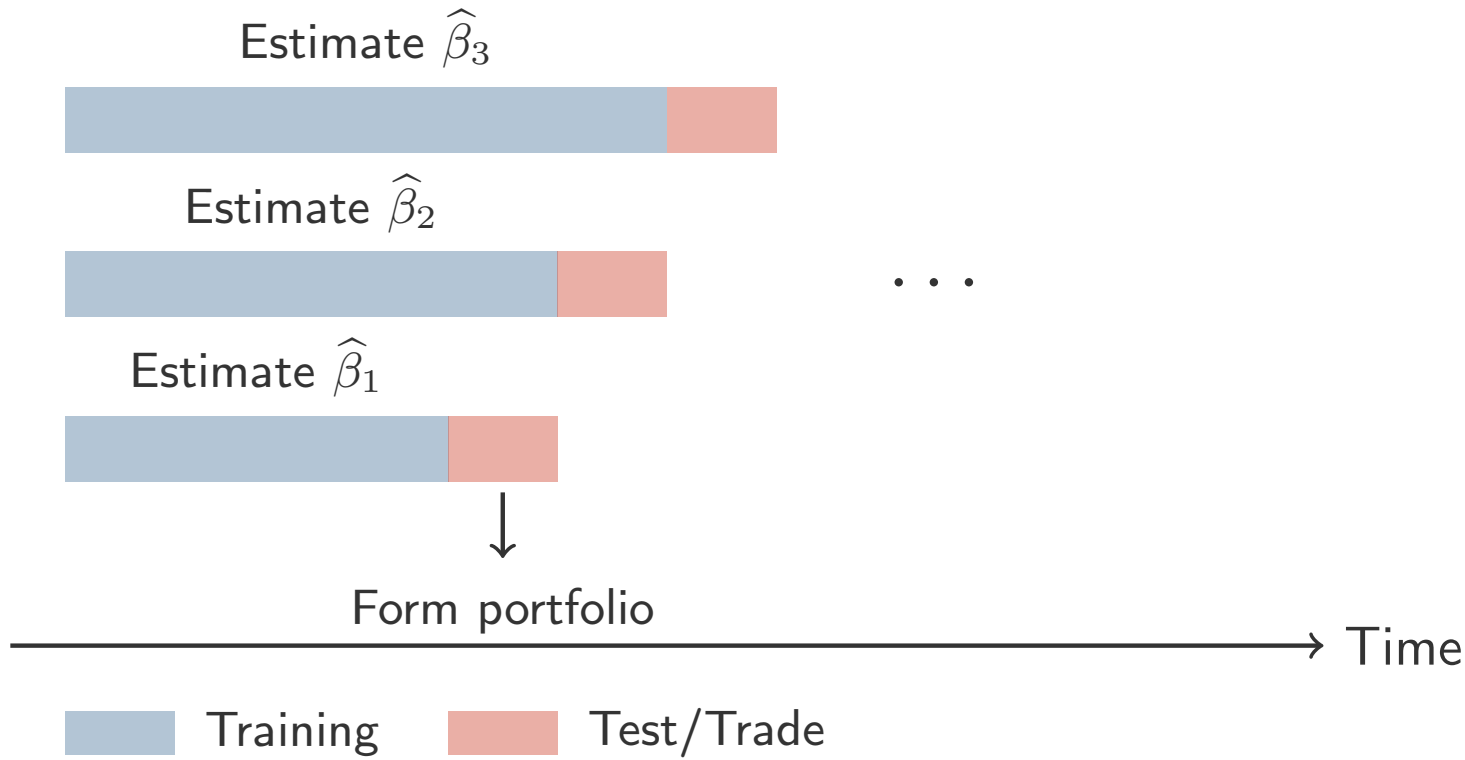
Procedure:

1. Estimate $\widehat{\beta}$ using data up to month t
2. For each stock i, compute predicted return:

$$\widehat{r}_{i,t+1} = x'_{i,t}\widehat{\boldsymbol{\beta}}$$

3. Rank stocks by $\widehat{r}_{i,t+1}$
4. Form portfolio:
   - Long: Top decile (highest predicted returns)
   - Short: Bottom decile (lowest predicted returns)
5. Observe realized portfolio return in month $t+1$
6. Roll forward, repeat

Walk-Forward Backtesting



Re-estimate periodically (monthly or quarterly).
Track cumulative returns.

Results: Shrinkage Helps Substantially

Kozak, Nagel & Santosh (2020) findings:

- Ridge substantially outperforms OLS out-of-sample
- Optimal λ implies significant shrinkage
  - Data supports skepticism about many characteristics
- Long-short portfolios achieve Sharpe ratios > 1
- Results robust across different sample periods
- Better performance than selecting ad-hoc subsets

Key message: Regularization is not just a technical fix — it embodies economically sensible skepticism.

Lessons from KNS (2020) for Practitioners

1. **Do not search for the "true" sparse model**
   - Financial markets are complex; many factors matter
   - Embrace high-dimensional methods rather than fight dimensionality

2. **Consider Ridge as baseline regularization method**
   - Especially when predictors are correlated (common in finance)
   - More robust than Lasso or OLS
   - Can always add L_1 penalty if you need interpretability

3. **Always validate out-of-sample**
   - In-sample fit is misleading with many predictors
   - Use proper time-series validation (walk-forward, cross-validation)
   - Test economic value, not just statistical fit

Some Key Practical Considerations

1. **Transaction costs**
   - ML strategies can have high turnover
   - Trading costs erode returns
   - May need to constrain turnover
2. **Capacity constraints**
   - ML strategies tend to work best in small/mid caps
   - Large trades move prices (market impact)
   - May not scale to institutional size (liquidity matters)
3. **Data quality**
   - Point-in-time data essential (avoid look-ahead)
   - Survivorship bias: Include delisted firms
   - Data errors can drive spurious results

Summary and Next Steps

Key Takeaways from Today

1. Market timing is very difficult with linear models — even regularization often helps only modestly
   - Monthly return volatility ≈ 4–5%
   - Predictable component $\approx 0.5\%$ (if any)
   - Signal buried in noise
   - The relationship between ERP and the state of the economy might be non-linear
2. Cross-sectional prediction benefits more clearly from shrinkage
   - Shrinkage = Skepticism about predictability
   - Economically sensible assumption given market timing evidence
3. Always evaluate the economic benefit of your model above and beyond statistical accuracy

Readings and Next Steps

Readings for Today's Lecture:

- Goyal, Welch, & Zafirov (2024). "A comprehensive 2022 look at the empirical performance of equity premium prediction." *The Review of Financial Studies*. *[Recomm.]*
- Kozak, Nagel & Santosh (2020), "Shrinking the Cross-Section," *Journal of Financial Economics* *[Supp.]*

Topics for the next lecture:

- Tree-based methods and ensemble learning
  - Random Forests: Bagging + feature sampling
  - Gradient Boosting: XGBoost, LightGBM
  - Non-linear relationships and interactions
- Key advantage: Automatically capture non-linearities and interactions that linear models miss.
- Can trees resurrect market timing and time-series predictability?

Questions?