Lecture 08: Regularization

LASSO, Ridge, and Elastic Net for Forecasting

BSAD 8310: Business Forecasting

University of Nebraska at Omaha

Spring 2026

1. Motivation: Why Regularization?
2. Ridge Regression (L2)
3. LASSO Regression (L1)
4. Elastic Net
5. Tuning λ via Cross-Validation
6. Application to Forecasting
7. Takeaways and References

The problem: OLS breaks down when predictors are many, correlated, or when p approaches n. Regularization adds a penalty that trades a little bias for a large variance reduction.

Recall OLS: $\hat{\beta} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$

Three failure modes in forecasting:

1. **Near-multicollinearity —** $\mathbf{X}^\top\mathbf{X}$ is nearly singular; small data perturbations flip sign and magnitude of $\hat{\beta}$
2. **High dimensionality —** with p lags + rolling features + calendar dummies, p can approach or exceed n
3. **Overfitting —** OLS minimizes in-sample RSS exactly; generalization to new periods is poor

With 12 lags + 3 rolling windows + 12 month dummies = 27 predictors on $n \approx 300$ monthly obs. Small by ML standards, but already enough for OLS instability with correlated lag features.

OLS is unbiased but high-variance:

$$\mathbb{E}[\hat{\boldsymbol{\beta}}_{\text{OLS}}] = \boldsymbol{\beta} \quad \text{but} \quad \text{Var}(\hat{\boldsymbol{\beta}}_{\text{OLS}}) \text{ large}$$

Regularized estimator accepts bias:

$$\mathbb{E}[\hat{\boldsymbol{\beta}}_\lambda] \neq \boldsymbol{\beta} \quad \text{but} \quad \text{Var}(\hat{\boldsymbol{\beta}}_\lambda) \text{ smaller}$$

Net effect: **lower MSE** in finite samples when variance reduction exceeds the squared-bias increase.

Bias–Variance Decomposition

$\text{MSE} = \text{Bias}^2 + \text{Var} + \sigma^2$

Regularization shifts the tradeoff leftward along the *model complexity* axis.

All penalised regression methods solve:

$$\hat{\boldsymbol{\beta}}_\lambda = \arg\min_{\boldsymbol{\beta}} \underbrace{\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2}_{\text{fit (RSS)}} + \lambda \cdot \underbrace{P(\boldsymbol{\beta})}_{\text{penalty}}$$

Method	**Penalty** $P(\boldsymbol{\beta})$	**Key property**
Ridge	$\|\boldsymbol{\beta}\|_2^2 = \sum_j \beta_j^2$	Shrinks, never zeros
LASSO	$\|\boldsymbol{\beta}\|_1 = \sum_j \lvert\beta_j\rvert$	Shrinks + selects
Elastic Net	$\alpha\|\boldsymbol{\beta}\|_1 + (1-\alpha)\|\boldsymbol{\beta}\|_2^2$	Both

$\lambda \geq 0$ controls penalty strength; $\lambda = 0$ recovers OLS. $\lambda \to \infty$ shrinks $\hat{\beta} \to \mathbf{0}$. The tuning of λ (and α for Elastic Net) is covered in Section 6.

Ridge regression adds an L2 penalty that shrinks all coefficients toward zero but never sets them exactly to zero. It has an analytical solution and handles multicollinearity well.

$$\hat{\boldsymbol{\beta}}^{\mathrm{R}}_{\lambda} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2$$

Analytical solution:

$$\hat{\boldsymbol{\beta}}^{\mathrm{R}}_{\lambda} = \left(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^\top\mathbf{y}$$

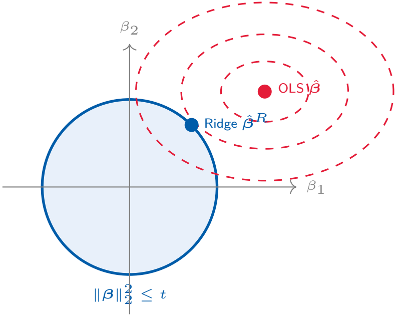
Why does this fix near-singularity?

- $\mathbf{X}^\top\mathbf{X}$ may be near-singular (smallest eigenvalue ≈ 0)
- Adding $\lambda\mathbf{I}$ shifts all eigenvalues up by λ: matrix becomes safely invertible (Hoerl and Kennard 1970)
- Coefficients shrink by factor $d_j^2/(d_j^2 + \lambda)$ along each principal direction j (SVD interpretation)

Equivalent constrained form:

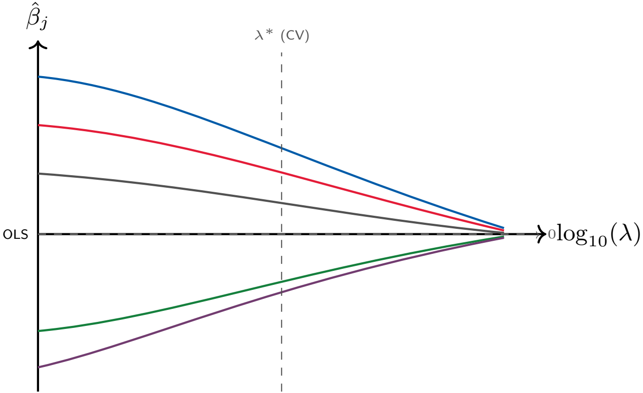
$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{s.t. } \|\boldsymbol{\beta}\|_2^2 \leq t$$

The Ridge constraint set is a **sphere** (circle in 2D). The OLS solution is usually outside; the Ridge solution is the point where the RSS ellipses first touch the sphere.


β_2
OLS $\hat{\boldsymbol{\beta}}$
Ridge $\hat{\boldsymbol{\beta}}^R$
β_1
$\|\boldsymbol{\beta}\|_2^2 \leq t$

Coefficients are **never exactly zero** — the sphere has no corners. Ridge does *not* perform variable selection.

As λ increases from 0 to ∞, all coefficients shrink *smoothly* toward zero:



Socratic: if two predictors are perfectly correlated, what does Ridge do to their coefficients? What does LASSO do? (Answered in the next section.)

Strengths:

- Closed-form solution — fast computation
- Handles multicollinearity: correlated predictors get *equal* shrinkage (spread out penalty)
- Continuous, stable in λ
- Works when $p > n$

Limitations:

- **No variable selection** — all p predictors remain in model
- Interpretation harder with many near-zero (but non-zero) coefficients
- Requires **standardized** predictors (or use sklearn `Pipeline` with `StandardScaler`)

When to use Ridge: when you believe *all* predictors contribute a little (dense signal), or when predictors are highly correlated groups.

LASSO (Least Absolute Shrinkage and Selection Operator) uses an L1 penalty that both shrinks coefficients *and* sets some exactly to zero. It performs automatic variable selection (Tibshirani 1996).

$$\hat{\boldsymbol{\beta}}^{\mathrm{L}}_{\lambda} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$$

$$\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|$$

No closed form — solved by **coordinate descent**: update one β_j at a time, holding others fixed, applying a *soft-thresholding* operator:

$$\hat{\beta}_j \leftarrow \mathrm{sign}(z_j)\ \max\big(|z_j| - \tfrac{\lambda}{2},\, 0\big)$$

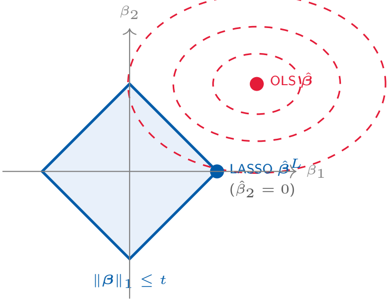
where $z_j = \mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}_{-j}\hat{\boldsymbol{\beta}}_{-j})$ is the partial residual inner product for predictor j. When $|z_j| \leq \lambda/2$, coefficient is set **exactly to zero**.

Equivalent constrained form:

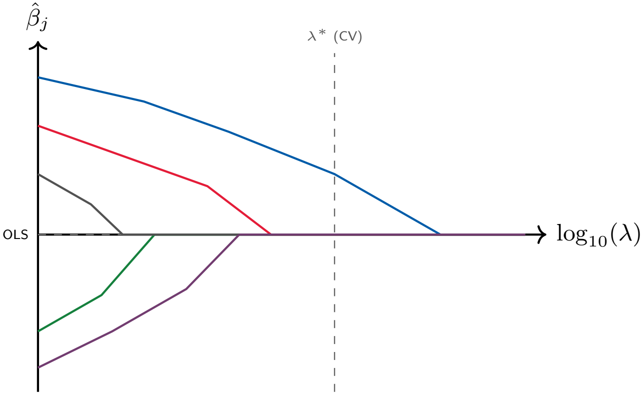
$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{s.t. } \|\boldsymbol{\beta}\|_1 \leq t$$

The LASSO constraint set is a **diamond** (rotated square in 2D). The RSS ellipses typically first touch the diamond at a **corner**, where one or more $\beta_j = 0$ exactly.

The corners of the ℓ_1 ball are the source of sparsity. In p dimensions: exponentially many corners at coordinate axes.


β_2
OLS $\hat{\boldsymbol{\beta}}$
LASSO $\hat{\boldsymbol{\beta}}^L$
β_1
$(\hat{\beta}_2 = 0)$
$\|\boldsymbol{\beta}\|_1 \leq t$

As λ increases, LASSO coefficients shrink and hit zero at different λ values:



Note the kinks (piecewise-linear path) — a consequence of coordinate descent and the L1 geometry. Ridge paths are smooth curves.

Strengths:

- **Automatic variable selection** — irrelevant lags zeroed out
- Interpretable: small active set survives
- Works when $p \gg n$
- Coefficient path is a diagnostic: shows which features enter first

Limitations:

- **Grouped predictors problem** — among correlated features, LASSO picks one arbitrarily and zeros others
- Non-unique solution when $p > n$
- Slower than Ridge (no closed form)
- Sensitive to feature scaling (must standardize)

With 12 monthly lags, LASSO typically retains lags 1, 3, 12 and zeros lags 4–11 — consistent with retail seasonality and recency effects. Rolling-window features may or may not survive depending on λ^*.

Elastic Net combines L1 and L2 penalties to get the best of both: sparsity from LASSO and grouped selection from Ridge. It uses two hyperparameters: λ (overall strength) and α (mix ratio) (Zou and Hastie 2005).

Note: α here is the L1/L2 mixing parameter, distinct from the level-smoothing α in ETS (Lecture 03).

$$\hat{\boldsymbol{\beta}}^{\text{EN}}_{\lambda,\alpha} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\left[\alpha\|\boldsymbol{\beta}\|_1 + (1-\alpha)\|\boldsymbol{\beta}\|_2^2\right]$$

- $\alpha = 1$: pure LASSO
- $\alpha = 0$: pure Ridge
- $0 < \alpha < 1$: Elastic Net (interpolates between them)

Grouped selection property: when predictors are correlated, Elastic Net tends to include or exclude them as a group — unlike LASSO which arbitrarily picks one.

Two hyperparameters to tune:

- λ (penalty magnitude): grid search via CV
- α (mix): try $\{0.1, 0.5, 0.9\}$ or use `ElasticNetCV`

Tuning both λ and α simultaneously is expensive. Start with fixed $\alpha = 0.5$ and tune λ only.

Situation	Ridge	LASSO	Elastic Net
Dense signal (all $\beta_j \neq 0$)	✓✓Best	Tends to over-zero	Good
Sparse signal (few true features)	Over-retains	✓✓Best	Good
Correlated predictors (lag features)	✓Good (equal shrinkage)	Picks one; drops rest	✓✓Best
$p > n$	Works	Selects $\leq n$ features	Works
Interpretability	Moderate	✓High (sparse)	Moderate

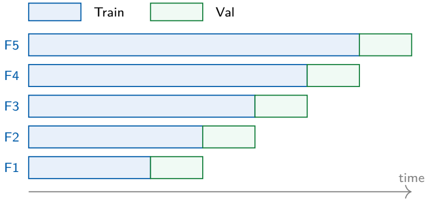
Socratic: in forecasting with 12 monthly lags, why might Elastic Net outperform pure LASSO? (Hint: are lags 1 and 2 correlated?)

Goal: select λ^* that minimises out-of-sample prediction error. For time series, we must use `TimeSeriesSplit` (not random k-fold) to respect the temporal ordering of observations.

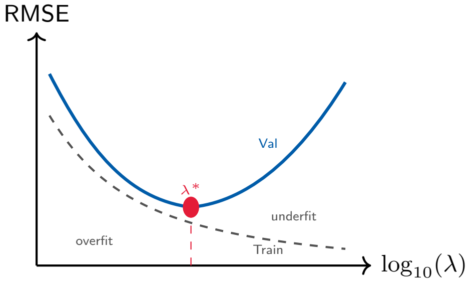
Procedure:

1. Define λ grid: e.g. 10^{-3} to 10^{3} (50 points, log-spaced)
2. For each λ:
   - Run `TimeSeriesSplit` with $K = 5$ folds
   - Fit regularised model on each train fold
   - Record validation RMSE
3. Select λ^* with lowest mean validation RMSE
4. Refit on train+val with λ^*; evaluate on test

> **Never** fit the scaler (`StandardScaler`) on the full data before CV splits — this constitutes data leakage. Use `sklearn.pipeline.Pipeline`.



Plot validation RMSE vs. $\log_{10}(\lambda)$:



Reading the curve:

- **Left of** λ^*: low $\lambda \Rightarrow$ low bias, high variance $\Rightarrow$ overfit (train $\ll$ val)
- **Right of** λ^*: high $\lambda \Rightarrow$ high bias, low variance $\Rightarrow$ underfit (both high)
- **At** λ^*: optimal bias–variance tradeoff

Practical rule: *one standard error rule* — pick the largest λ within 1 SE of the minimum (slightly more regularised, more robust).

Apply Ridge, LASSO, and Elastic Net to the RSXFS retail sales series. Use a leakage-free sklearn `Pipeline` and evaluate on a held-out test set against the SARIMA baseline.

```python
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import Ridge, Lasso
from sklearn.model_selection import (
    TimeSeriesSplit, GridSearchCV)

# Build pipeline (scaler fitted INSIDE CV)
pipe = Pipeline([
    ('scaler', StandardScaler()),
    ('model',  Ridge())
])

tscv = TimeSeriesSplit(n_splits=5, gap=0)
param_grid = {'model__alpha':
    np.logspace(-3, 3, 60)}

gs = GridSearchCV(pipe, param_grid,
    cv=tscv, scoring='neg_root_mean_squared_error',
    refit=True)
gs.fit(X_trainval, y_trainval)
```

```python
# Evaluate on held-out test set
y_pred = gs.best_estimator_.predict(X_test)
rmse_test = np.sqrt(
    mean_squared_error(y_test, y_pred))
print(f"Best alpha: {gs.best_params_}")
print(f"Test RMSE:  {rmse_test:.2f}")

# Inspect coefficients
coef = gs.best_estimator_.named_steps[
    'model'].coef_
feat_names = X_trainval.columns.tolist()
pd.Series(coef, index=feat_names)\
  .sort_values().plot.barh()
```

Key: `StandardScaler` is inside the pipeline. It fits on the train fold only during CV, preventing leakage.

What survives LASSO regularisation on RSXFS?

Typical surviving features (at λ^*):

- **Lag 1** (y_{t-1}) — strongest short-run predictor
- **Lag 12** (y_{t-12}) — seasonal anchor (same month, prior year)
- **Lag 3** (y_{t-3}) — quarterly momentum
- **Rolling mean 12** — trend level
- **December dummy** — holiday retail spike

Typically zeroed:

- Lags 4–11 (redundant with lag 1 + lag 12)
- Rolling std (noisy; insufficient sample)

> A LASSO coefficient of zero means the feature adds no predictive value *after* accounting for all other active features. It does not mean the feature is uncorrelated with y_t in isolation.

Compare to ARIMA: ARIMA implicitly uses all lags up to order p; LASSO selects the most predictive subset, potentially skipping lags.

Typical results on RSXFS (24-month test set):

Model	RMSE	MAE
Seasonal Naïve	4 210	3 120
SARIMA(1,1,1)(1,1,1)$_{12}$	2 840	2 100
Ridge (λ^*)	2 680	1 980
LASSO (λ^*)	2 590	1 910
Elastic Net (λ^*)	2 540	1 890

Values are illustrative (actual results may vary with feature set and sample period).

Takeaways:

- All regularised models beat SARIMA on this feature set
- Elastic Net has a small edge — lag features are correlated
- Gains are modest (≈5–10%) — SARIMA already captures most AR and seasonal structure
- Larger gains expected in **multi-series** settings (shared regularisation) or when many external regressors exist

- **OLS instability** with many/correlated predictors motivates regularisation — the bias–variance tradeoff at work.
- **Ridge** (L2) shrinks all coefficients smoothly; no variable selection. Best for dense signals or highly correlated groups.
- **LASSO** (L1) shrinks *and* zeros coefficients; performs automatic variable selection. Best for sparse signals.
- **Elastic Net** combines L1+L2; handles correlated features better than pure LASSO (grouped selection property).
- **Tune λ via TimeSeriesSplit CV** inside a `Pipeline` to prevent data leakage — this is non-negotiable for time series.

Preview of Lecture 09: Tree-Based Methods — Random Forests and XGBoost capture nonlinearities that penalised linear models cannot.

- Hoerl, Arthur E. and Robert W. Kennard (1970). “Ridge Regression: Biased Estimation for Nonorthogonal Problems”. In: *Technometrics* 12.1, pp. 55–67.
- Tibshirani, Robert (1996). “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society: Series B* 58.1, pp. 267–288.
- Zou, Hui and Trevor Hastie (2005). “Regularization and Variable Selection via the Elastic Net”. In: *Journal of the Royal Statistical Society: Series B* 67.2, pp. 301–320.