Ridge Regression & LASSO

Regularization for High-Dimensional Economics

L2 shrinkage · L1 selection · Elastic Net · Bias-Variance · Cross-Validation

Instructor	Course	Prerequisite	Language
Assoc. Prof. Dr. Onur Polat	ECO447 — Core (B.Sc.)	Weeks 1–5 completed	English

Week 6 — Agenda

What we will cover today

01 The Curse of Dimensionality

02 Ridge Regression (L2)

03 LASSO Regression (L1)

04 Elastic Net

05 Choosing λ by Cross-Validation

06 Economic Applications

07 Full Pipeline in Python

08 Summary and Next Week

From OLS and Logit to Regularization

Why we need to go beyond classical methods

What we have

- Weeks 4-5: OLS, Logistic Regression
- Classical p < n, well-conditioned X
- Interpretable coefficients
- Bias ≈ 0, variance manageable

The new challenge

- Modern economic data: p can exceed n
- Macro: 100+ monthly indicators
- Finance: 1000+ firm characteristics
- OLS fails when X'X is singular

Today's tools

- Ridge (L2): shrink all coefficients
- LASSO (L1): shrink AND select
- Elastic Net: combine both penalties
- Pick strength (λ) via cross-validation

Key insight

- Accept small BIAS → large VARIANCE drop
- Total prediction error FALLS
- Standardize features FIRST (scale-sensitive)
- Interpretability + prediction accuracy

The Curse of Dimensionality

When classical regression breaks down

Why Regularization? The p ≥ n Problem

When OLS fails and bias becomes our friend

- Modern economic data: p (features) can equal or exceed n (observations)
- Macroeconomic forecasting: hundreds of monthly predictors for GDP, inflation
- Financial data: thousands of firm characteristics predict stock returns
- OLS fails when p ≥ n: the matrix X'X is singular, so $\beta = (X'X)^{-1}X'y$ is undefined
- Even when p < n: many irrelevant variables cause overfitting and high variance
- Regularization adds a PENALTY term to the loss to shrink or zero out coefficients
- Key insight: accept some bias to gain a LARGE reduction in variance

OLS vs Regularized Regression

The fundamental trade-off

OLS (No Regularization)

- Minimizes: $\Sigma (y_i - x_i'\beta)^2$
- Unbiased coefficients (Gauss-Markov BLUE)
- High variance when p is large
- Includes all variables equally
- No feature selection
- Fails when $p \geq n$ (singular $X'X$)
- Sensitive to multicollinearity

Regularized Regression

- Minimizes: $\Sigma (y_i - x_i'\beta)^2 + \lambda \cdot \text{Penalty}(\beta)$
- BIASED but lower variance
- Stable even with many features
- LASSO: automatic feature selection
- Works when $p >> n$
- Shrinks unimportant coefficients
- λ controls the regularization strength

The Bias-Variance Trade-off

The theoretical motivation for regularization

$$E[(y - \hat{f}(x))^2] = \text{Bias}^2 + \text{Variance} + \sigma^2$$

- OLS is UNBIASED but can have HIGH variance when p is large
- Regularization INTRODUCES BIAS but DRAMATICALLY REDUCES variance
- Total error can be much lower than OLS → better out-of-sample prediction
- As λ increases: Bias ↑, Variance ↓
- λ is a hyperparameter: tuned via cross-validation (not fit from data)
- This is a form of "shrinkage" — pulling coefficients toward zero

Ridge Regression (L2)

Shrinking coefficients with a squared penalty

Ridge Regression — Mathematical Formulation

The L2 penalty

$$\beta_Ridge = \text{argmin}_\beta \{ \Sigma_i (y_i - x_i'\beta)^2 + \lambda \Sigma_j \beta_j^2 \}$$

- Adds the squared coefficients (L2 norm) as a penalty — hence "L2 regularization"
- $\lambda \geq 0$ controls the strength: $\lambda = 0$ recovers OLS, $\lambda \to \infty$ forces all $\beta_j \to 0$
- Closed-form solution: $\beta_Ridge = (X'X + \lambda I)^{-1} X'y$ — always invertible since λI makes $X'X + \lambda I$ positive definite
- Shrinks coefficients CONTINUOUSLY toward zero, but never exactly zero
- Equivalent to a Bayesian regression with a Gaussian prior $N(0, \sigma^2/\lambda)$ on β

Ridge Regression — Properties

What Ridge is good for, and what it cannot do

Strengths

- Handles multicollinearity
- Stabilizes coefficient estimates
- Works even when p > n
- Closed-form (very fast)
- Lower variance than OLS
- All features kept — no information lost

Limitations

- NO variable selection
- All p features remain in model
- Interpretation harder (many small coefs)
- Not suitable when goal is sparsity
- Still biased (not BLUE)
- Must standardize features

Economic use cases

- Macro forecasting (many lags)
- Portfolio optimization (prevents extreme weights)
- Hedonic price regressions
- Hundreds of indicator variables
- Correlated factors (e.g. industry dummies)
- Prior-regularized Bayesian setup

Ridge Regression in Python

scikit-learn + cross-validated λ selection

PYTHON

```
import numpy as np
from sklearn.linear_model import Ridge, RidgeCV
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

# ALWAYS standardize before Ridge — penalty is scale-sensitive
pipe_ridge = Pipeline([
    ('scaler', StandardScaler()),
    ('ridge',  RidgeCV(alphas=np.logspace(-4, 4, 100),
                       cv=5, scoring='neg_mean_squared_error')),
])

pipe_ridge.fit(X_train, y_train)
print(f"Best alpha: {pipe_ridge['ridge'].alpha_:.4f}")
print(f"Test R²:    {pipe_ridge.score(X_test, y_test):.4f}")

# Coefficients
coefs = pd.Series(pipe_ridge['ridge'].coef_,
                  index=X_train.columns).sort_values(key=abs, ascending=False)
print(coefs.head(10))
```

LASSO Regression (L1)

Shrinkage + automatic variable selection

LASSO (Tibshirani, 1996)

Least Absolute Shrinkage and Selection Operator

$$\beta_LASSO = \text{argmin}_\beta \{ \Sigma_i (y_i - x_i'\beta)^2 + \lambda \Sigma_j |\beta_j| \}$$

- Uses the L1 norm (absolute values) instead of the L2 (squared) penalty
- No closed-form solution — solved by coordinate descent, LARS, or proximal gradient
- Soft-thresholding (orthogonal case): $\beta_j = \text{sign}(\beta_j_OLS) \cdot \max(|\beta_j_OLS| - \lambda, 0)$
- KEY PROPERTY: LASSO sets some coefficients EXACTLY to zero
- $\lambda = 0$: recovers OLS; $\lambda \to \infty$: all β_j become 0
- Combines shrinkage AND variable selection in a single step

Why Does LASSO Set Coefficients to Zero?

The geometric intuition

The L1 ball has corners

- Minimize RSS subject to $\Sigma|\beta_j| \leq t$
- The constraint region is a diamond (L1 ball)
- Its corners sit ON the axes → $\beta_j = 0$
- OLS contours (ellipses) are likely to touch the diamond at a corner
- At a corner, one or more coefficients are exactly 0
- By contrast, the Ridge disk is smooth → coefficients only shrink

Practical consequences

- LASSO produces SPARSE solutions — many β_j exactly 0
- Automatic variable selection — model + selection in one step
- Interpretable: a short, selected list of predictors
- Economically appealing: which of 100+ predictors truly matter?
- Caution: with correlated features, LASSO ARBITRARILY picks one
- Post-LASSO: refit OLS on selected variables to remove shrinkage bias

LASSO in Python

Automatic feature selection with LassoCV

PYTHON

```
from sklearn.linear_model import LassoCV
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

pipe_lasso = Pipeline([
    ('scaler', StandardScaler()),
    ('lasso',  LassoCV(cv=5, random_state=42,
                       max_iter=10000,
                       n_alphas=100)),
])
pipe_lasso.fit(X_train, y_train)

best_alpha = pipe_lasso['lasso'].alpha_
coefs = pipe_lasso['lasso'].coef_
n_selected = (coefs != 0).sum()
print(f"Best alpha:        {best_alpha:.6f}")
print(f"Features selected: {n_selected} / {X_train.shape[1]}")

# Show selected features (non-zero coefficients)
sel = pd.Series(coefs, index=X_train.columns)
sel = sel[sel != 0].sort_values(key=abs, ascending=False)
print(sel)
```

Elastic Net

Best of both worlds — L1 selection + L2 grouping

Elastic Net (Zou & Hastie, 2005)

A convex combination of L1 and L2

$$\beta_{EN} = \text{argmin}_\beta \{ RSS + \lambda [\alpha \Sigma_j |\beta_j| + (1-\alpha)/2 \cdot \Sigma_j \beta_j^2] \}$$

- Mixing parameter $\alpha \in [0, 1]$ controls the blend: $\alpha = 1 \rightarrow$ pure LASSO, $\alpha = 0 \rightarrow$ pure Ridge
- Pure LASSO can be unstable when predictors are strongly correlated
- The L2 part encourages a GROUPING effect: correlated predictors enter or leave together
- The L1 part preserves variable selection (sparsity)
- In practice: cross-validate over BOTH α AND λ (e.g. $\alpha \in \{0.1, 0.5, 0.9\}$)
- Often outperforms both pure Ridge and pure LASSO on real economic data

Ridge vs LASSO vs Elastic Net — Summary

Which should you use when?

Ridge (L2)

- Penalty: $\Sigma\beta_j^2$
- Shrinks ALL coefficients
- No variable selection
- Closed-form solution
- Great for multicollinearity
- All p features retained
- Best when: all features matter

LASSO (L1)

- Penalty: $\Sigma|\beta_j|$
- Shrinks + ZEROS OUT coefs
- Automatic variable selection
- Iterative (coordinate descent)
- Unstable with correlated X
- Sparse: subset of features
- Best when: few predictors truly matter

Elastic Net

- Penalty: α L1 + (1-α) L2
- Shrinks + selects
- Groups correlated features
- Iterative (coordinate descent)
- Robust to correlated predictors
- α and λ via CV
- Best when: unsure — safe default

Choosing λ by Cross-Validation

How to tune the regularization strength

K-Fold Cross-Validation for λ

Why λ cannot be estimated from training data alone

- λ is a HYPERPARAMETER: not a model coefficient — training alone always prefers λ = 0 (OLS)
- Use K-fold cross-validation (typically K = 5 or 10) to pick λ
- Grid of λ values on a log scale: e.g. np.logspace(-4, 4, 100)
- For each λ: fit on K-1 folds, measure MSE on held-out fold, average across folds
- Pick the λ that minimizes mean CV error (λ_min) — best predictive performance
- OR pick the "1-SE rule": largest λ within 1 SE of λ_min — more parsimonious model
- Information criteria (AIC, BIC) are alternatives, but CV is the ML standard

λ_min vs λ_1se — A Common Choice

Accuracy vs parsimony

λ_min — Minimum CV Error

- The λ that MINIMIZES cross-validation MSE
- Best predictive performance on average
- Tends to include more features (less sparse)
- Default for sklearn's LassoCV / RidgeCV
- Use when: prediction accuracy is paramount

λ_1se — One-Standard-Error Rule

- Largest λ within 1 SE of λ_min
- MORE regularization → simpler, sparser model
- Trades a tiny bit of accuracy for interpretability
- Popular in econometrics (Hastie, Tibshirani)
- Use when: parsimony and interpretation matter

Visualizing the Regularization Path

How coefficients shrink and vanish as λ grows

PYTHON

```
from sklearn.linear_model import lasso_path
import matplotlib.pyplot as plt

alphas, coefs, _ = lasso_path(X_train_std, y_train,
                              alphas=np.logspace(-4, 1, 100))

fig, ax = plt.subplots(figsize=(10, 5))
for i, name in enumerate(X_train.columns):
    ax.plot(np.log10(alphas), coefs[i], label=name)
ax.axvline(np.log10(lasso_cv.alpha_), ls='--', color='k',
           label=f'λ_min = {lasso_cv.alpha_:.4f}')
ax.set_xlabel('log₁₀(λ)')
ax.set_ylabel('Coefficient value')
ax.set_title('LASSO Regularization Path')
ax.legend(bbox_to_anchor=(1.02, 1), fontsize=8)
plt.tight_layout(); plt.show()

# Interpretation: features that survive at higher λ are the
# most robust predictors; those that drop out early are weaker.
```

Economic Applications

Macro forecasting · Asset pricing · Policy prediction

LASSO in Macroeconomic Forecasting

Which of hundreds of indicators actually matter?

- GDP nowcasting: select the most predictive from 100+ monthly / weekly indicators
- Inflation forecasting: which sub-indices of CPI really drive aggregate inflation?
- Phillips curve: select the relevant slack measures among many candidates (U, employment gap, etc.)
- Factor-augmented LASSO: combine PCA factors with LASSO selection on individual series
- Stock & Watson (2012): LASSO vs PCA factors for macro forecasting — comparable accuracy, better interpretability
- LASSO-selected variables often align with economic theory → bridge ML and classical econometrics

Regularization in Finance

Asset pricing, portfolios, factor selection

- Asset pricing: which of 100+ firm characteristics predict cross-sectional returns?
- Portfolio optimization: Ridge prevents extreme weights (shrinks toward equal-weight)
- Risk factor selection: LASSO identifies the small set of relevant risk factors
- Fama-MacBeth with LASSO: regularized cross-sectional regressions
- Gu, Kelly & Xiu (JFE 2020): Elastic Net is one of the top ML asset-pricing methods
- Regularization as an "economic prior": uncertain effects → shrink toward zero
- LASSO for instrument selection in IV regression (Belloni, Chernozhukov, Hansen)

Full Pipeline in Python

End-to-end: data → scaling → CV → comparison

Comparing OLS · Ridge · LASSO · Elastic Net

A reproducible pipeline

PYTHON

```
from sklearn.linear_model import (LinearRegression, RidgeCV,
                                  LassoCV, ElasticNetCV)
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.metrics import r2_score, mean_squared_error

models = {
    'OLS':         LinearRegression(),
    'Ridge':       RidgeCV(alphas=np.logspace(-4, 4, 100), cv=5),
    'LASSO':       LassoCV(cv=5, max_iter=10000, random_state=42),
    'ElasticNet':  ElasticNetCV(l1_ratio=[0.1, 0.5, 0.7, 0.9, 0.95],
                                cv=5, max_iter=10000, random_state=42),
}

results = []
for name, model in models.items():
    pipe = Pipeline([('sc', StandardScaler()), ('m', model)])
    pipe.fit(X_train, y_train)
    results.append({'Model': name,
                    'Test R²': pipe.score(X_test, y_test),
                    'Test RMSE': np.sqrt(mean_squared_error(y_test,
                                         pipe.predict(X_test))),
                    'Non-zero β': int((pipe['m'].coef_ != 0).sum())})
print(pd.DataFrame(results).round(4).to_string(index=False))
```

Common Pitfalls

What goes wrong in practice

Mistake	Fix
• Not standardizing features before fit	• ALWAYS: StandardScaler before Ridge/LASSO
• Using the test set to tune λ	• Use nested CV or hold out a clean test set
• Reporting R^2 on the training set	• Report test / cross-validated R^2
• Interpreting LASSO signs under multicollinearity	• Use Elastic Net or group-LASSO instead
• Using λ = 0.5 or some default without CV	• Grid-search λ via LassoCV / RidgeCV
• Ignoring 1-SE rule when parsimony matters	• Compare both and report simpler one if close

Week 6 — Key Takeaways

What to carry into the midterm and beyond

- Regularization trades a small amount of BIAS for a large reduction in VARIANCE
- Essential when p is large relative to n — classical OLS fails or overfits
- RIDGE (L2): shrinks ALL coefficients continuously, handles multicollinearity, keeps all features
- LASSO (L1): shrinks AND SELECTS — sets some coefficients exactly to zero
- ELASTIC NET: a blend of L1 and L2 — best default for correlated predictors
- λ chosen by CROSS-VALIDATION: λ_min for accuracy, λ_1se for parsimony
- STANDARDIZE features FIRST — all three penalties are scale-sensitive
- After the midterm: Decision Trees — a fundamentally different, non-linear approach

References and Next Week

Further reading and the road ahead

Essential references

- Hastie, Tibshirani, Friedman (2009), ESL — Chapter 3.4
- James, Witten, Hastie, Tibshirani (2021/2023), ISLR — Chapter 6
- Tibshirani (1996), "Regression Shrinkage and Selection via LASSO", JRSS-B
- Zou & Hastie (2005), "Regularization and Variable Selection via the Elastic Net", JRSS-B
- Belloni, Chernozhukov, Hansen (2014), "High-Dimensional Methods in Econometrics", JEP
- scikit-learn docs: linear_model.LassoCV, RidgeCV, ElasticNetCV

After the midterm

- Week 7: Decision Trees — a non-linear, interpretable alternative
- Recursive partitioning: splits on features, no coefficients
- Handles non-linearities and interactions automatically
- Natural handling of mixed data types
- Foundation for ensemble methods (Random Forest, Boosting)
- Classical reference: Breiman et al. (1984), CART