Lesson 22 Summary: Regularization

Data Science with Python – Key Concepts

Data Science Program

Regularization Overview

Regularization

Ridge (L2)

Shrinks coefficients
Keeps all features
Circular constraint

Lasso (L1)

Zeroes coefficients
Feature selection
Diamond constraint

ElasticNet

Combines L1+L2
Best of both
Two hyperparams

Lambda Selection

Cross-validation (CV)
GridSearchCV for optimal alpha

Overfitting Prevention

Bias-variance tradeoff
Simpler models generalize better

Ridge(alpha=1.0) | Lasso(alpha=1.0) | ElasticNet(alpha=1.0, l1_ratio=0.5)

Regularization prevents overfitting by constraining model complexity

Why Regularize?

The overfitting problem:

- **Symptom:** Great training fit, poor test performance
- **Cause:** Model memorizes noise, not signal
- **Solution:** Penalize large coefficients

Key principle:
Simpler models generalize better to new data.

Regularization trades bias for reduced variance

Ridge Regression (L2)

Add squared penalty:

- **Penalty:** $\lambda \sum w_i^2$
- **Effect:** Shrinks all coefficients toward zero
- **Geometry:** Circular constraint region

Key property:
Keeps all features, just makes coefficients smaller.

Ridge works well with correlated features

Lasso Regression (L1)

Add absolute value penalty:

- **Penalty:** $\lambda \sum |w_i|$
- **Effect:** Sets some coefficients exactly to zero
- **Geometry:** Diamond constraint region

Key property:
Performs automatic feature selection.

Lasso produces sparse, interpretable models

ElasticNet

Combines L1 and L2 penalties:

- **Two parameters:** alpha (strength), l1_ratio (mix)
- **l1_ratio=0:** Pure Ridge
- **l1_ratio=1:** Pure Lasso

Best of both worlds:
Feature selection + grouped feature handling.

ElasticNet handles correlated features better than Lasso

Choosing Lambda

Cross-validation for hyperparameter tuning:

- **GridSearchCV:** Try many alpha values
- **RidgeCV/LassoCV:** Built-in CV versions
- **Metric:** Usually minimize MSE

Always use cross-validation to select regularization strength

Feature Scaling

Critical for regularization:

- **Problem:** Different scales = different penalties
- **Solution:** StandardScaler before fitting
- **Result:** Fair comparison across features

Always scale features before regularized regression!

Regularization penalizes coefficient magnitude, not importance

Finance Application

Factor model selection:

- **Many potential factors:** Which ones matter?
- **Lasso:** Identifies relevant factors
- **Ridge:** Stable estimates with correlated factors

Regularization improves out-of-sample prediction

Ridge vs Lasso Decision

When to use each:

- **Ridge:** Many small effects, correlated features
- **Lasso:** Few important features, need selection
- **ElasticNet:** Uncertain, or both apply

Start with ElasticNet, then specialize

Quick Reference

Essential Commands:

Method	Code
Ridge	`Ridge(alpha=1.0).fit(X, y)`
Lasso	`Lasso(alpha=1.0).fit(X, y)`
ElasticNet	`ElasticNet(alpha=1.0, l1_ratio=0.5)`
CV tuning	`RidgeCV(alphas=[...])`
Scale data	`StandardScaler().fit_transform(X)`

Regularization is essential for high-dimensional data