Regularization

Shreeyesh Menon

1 Regression with Standardized Features

Given a target variable y and features (f_1, f_2), we first consider the linear regression model with an intercept:

$$y = \beta_0 + \beta_1 f_1 + \beta_2 f_2 + \varepsilon$$

The OLS estimate of the parameter vector is:

$$\boldsymbol{\beta}_{\text{OLS}} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

If we standardize y, f_1, and f_2 (demeaning and dividing by their standard deviations), we remove the need for an intercept:

$$\tilde{y} = \beta_1 \tilde{f}_1 + \beta_2 \tilde{f}_2 + \varepsilon$$

The OLS estimator for the standardized regression becomes:

$$\boldsymbol{\beta}_{\text{OLS}} = (\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top \tilde{y}$$

2 Loss Function Schematic

We compare the objective functions minimized by the three estimators:

Estimator	**Loss Function**
OLS	$\min_\beta \lVert \tilde{y} - \tilde{X}\beta \rVert_2^2$
Ridge	$\min_\beta \left\{ \lVert \tilde{y} - \tilde{X}\beta \rVert_2^2 + \lambda \lVert \beta \rVert_2^2 \right\}$
LASSO	$\min_\beta \left\{ \lVert \tilde{y} - \tilde{X}\beta \rVert_2^2 + \lambda \lVert \beta \rVert_1 \right\}$

2.1 Features:

All three aim to reduce the sum of squared residuals.

- OLS minimizes the in-sample mean-squared error, but does not necessarily generalize well out-of-sample
- Ridge adds an ℓ_2 penalty to shrink coefficients: the coefficients are brought closer to the origin in $\beta-$space
- LASSO adds an ℓ_1 penalty that encourages sparsity. Some coefficients are dropped entirely: variable selection or *dropout* is happening here.

2. Ridge Regression

Ridge regression adds an ℓ_2 penalty to shrink coefficients:

$$\boldsymbol{\beta}_{\text{ridge}} = (\tilde{X}^\top \tilde{X} + \lambda I)^{-1} \tilde{X}^\top \tilde{y}$$

- $\lambda \geq 0$ controls the strength of the penalty. - Shrinks coefficients toward zero but does not set them exactly to zero. - Useful in high-dimensional settings or when predictors are correlated.

3. LASSO Regression

LASSO adds an ℓ_1 penalty to encourage sparsity:

$$\boldsymbol{\beta}_{\text{LASSO}} = \arg\min_{\beta} \left\{ \|\tilde{y} - \tilde{X}\beta\|_2^2 + \lambda \|\beta\|_1 \right\}$$

- Can set some coefficients exactly to zero. - Provides automatic feature selection.

Why LASSO Creates Zero Coefficients

- The constraint region of LASSO is a diamond (in 2D), with corners on the axes. - The loss function's level curves often touch the constraint region at these corners. - Hence, LASSO solutions often lie on the axes, setting coefficients exactly to zero. - In contrast, ridge has a round constraint region, which rarely touches the axes.

4. Training MSE Comparison

$$\text{MSE}_{\text{train}}^{\text{OLS}} \leq \text{MSE}_{\text{train}}^{\text{ridge}} \leq \text{MSE}_{\text{train}}^{\text{LASSO}}$$

- OLS fits the training data best (lowest training MSE). - Ridge and LASSO trade off training error for better generalization.

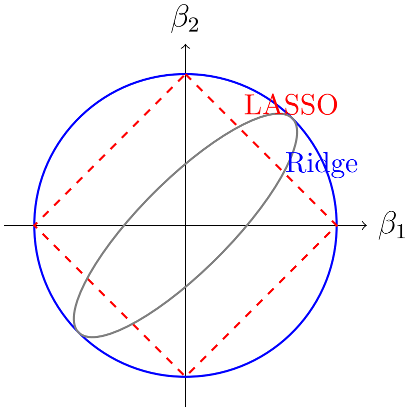
3 LASSO vs. Ridge: Visual Intuition

Key idea: The difference lies in the *geometry* of their constraint regions.

Constraint Shapes

- **Ridge regression** imposes an ℓ_2 constraint: $\|\beta\|_2^2 \leq c$.
- **LASSO** imposes an ℓ_1 constraint: $\|\beta\|_1 \leq c$.

Resulting shapes in 2D:



What Happens?

- The ellipses are contours of the loss function.
- The solution is found where a contour first touches the constraint region.
- For **ridge**, this is usually a point on the edge of the circle — unlikely to lie on the axes.
- For **LASSO**, the corners of the diamond lie on the axes — making it likely that the optimal solution sets some $\beta_j = 0$.

Mathematical Intuition: Ridge and LASSO introduce additional constraints by augmenting the loss function with L1 and L2 penalties on coefficients. These introduce a marginal benefit from shrinking the parameters. However, since L2 involves a quadratic, the marginal benefit from shrinkage dies out as the coefficients get smaller, so the model stops short of setting coefficients all the way to zero. In the LASSO case, the marginal benefit from shrinkage is constant. Hence, the model finds it optimal to keep shrinking until one or more of the coefficients is set exactly to zero.

4 Practitioner's guide

Selecting λ via Cross-Validation

To choose the optimal regularization parameter λ for ridge or LASSO, use K-fold cross-validation:

1. Select a grid of λ values (e.g., $\lambda \in \{10^{-4}, 10^{-3}, \ldots, 10^{2}\}$).
2. Split the data into K folds (commonly $K = 5$ or 10).
3. For each λ, train the model on $K - 1$ folds and compute the validation MSE on the remaining fold.
4. Average the validation MSE across all folds.
5. Choose the λ with the lowest average validation error.
6. Retrain the model on the full dataset using this optimal λ.

Note: This procedure helps balance bias and variance, improving generalization to unseen data.

Python Example: Cross-Validation to Select λ

Below is a Python example using `scikit-learn` to select λ for Ridge and LASSO using cross-validation.

This script performs 5-fold cross-validation and prints the best λ for each model. """

A Note on Categorical Variables

Categorical variables can be incorporated into regression models via **one-hot encoding**, where each category becomes a separate binary feature.

Example: A variable `color` with values {red, green, blue} can be encoded as:

`color`	`color_red`	`color_green`
red	1	0
green	0	1
blue	0	0

The baseline category (here, `blue`) is omitted to avoid multicollinearity.

Should One-Hot Encoded Variables Be Standardized?

No. One-hot encoded features take values in {0, 1} and do not require standardization. Scaling them can distort the meaning of the coefficients, giving dubious results.

Regularization and One-Hot Encoded Variables

Ridge Regression

- Applies an ℓ_2 penalty to all coefficients. - Shrinks dummy variables but retains all of them. - Useful when many categories are present and overfitting is a concern.

LASSO Regression

- Applies an ℓ_1 penalty. - May shrink some dummy coefficients to exactly zero, effectively dropping specific levels. - Can cause interpretational issues if only some levels of a categorical variable remain.

Group LASSO

- Groups together the dummy variables for each original categorical variable. - Applies penalty at the group level, allowing all levels of a categorical variable to be dropped together. - Maintains interpretability.

4. Summary Table

Method	**Keeps All Dummies**	**Can Drop Some**	**Can Drop Entire Variable**
OLS	Yes	No	No
Ridge	Yes (shrunk)	No	No
LASSO	Not necessarily	Yes	No
Group LASSO	No	No	Yes

5 Group Lasso in Python

```
import numpy as np
from group_lasso import GroupLasso

# Generate sample data
X = np.random.rand(100, 10)
y = np.random.rand(100)

# Define group structure (e.g., 2 groups of 5 features each)
groups = np.array([1, 1, 1, 1, 1, 2, 2, 2, 2, 2])

# Initialize and fit Group Lasso model
model = GroupLasso(groups=groups, group_reg=0.1, l1_reg=0)
model.fit(X, y)
```

```
# Get selected features (groups)
selected_groups = np.unique(groups[model.coef_ != 0])

print("Selected groups:", selected_groups)
```