Lasso Regression

Noted by Md. Inzamamul Lohani

Software Engineering, IIT, DU

November 22, 2026

Contents

1 The Big Picture

1.1 What is Lasso Regression?

Lasso stands for **Least Absolute Shrinkage and Selection Operator**. It is a regularized linear regression method. Like ordinary least squares (OLS), it tries to predict a response variable from explanatory variables. Unlike OLS, it adds a penalty that discourages large coefficients and can make some coefficients exactly zero.

For n observations and p predictors, write the linear model as

$$y = X\beta + \varepsilon, \tag{1.1}$$

where $y \in \mathbb{R}^{n\times 1}$ is the response vector, $X \in \mathbb{R}^{n\times p}$ is the design matrix, $\beta \in \mathbb{R}^{p\times 1}$ is the coefficient vector, and ε is the error vector. An intercept can be included separately.

1.2 Why do we need regularization?

OLS minimizes the residual sum of squares:

$$\hat{\beta}_{\text{OLS}} = \arg\min_{\beta} \|y - X\beta\|_2^2 . \tag{1.2}$$

When predictors are numerous, noisy, or strongly related to one another, an OLS model may have unstable coefficients or overfit the training data. Regularization adds a penalty to the fitting objective to control coefficient size.

1.2.1 Ridge versus Lasso

Ridge uses the squared L_2 penalty:

$$J_{\text{Ridge}}(\beta) = \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} \beta_j^2 . \tag{1.3}$$

Lasso uses the L_1 penalty:

$$\boxed{J_{\text{Lasso}}(\beta) = \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j| .} \tag{1.4}$$

The parameter $\lambda \geq 0$ controls the strength of regularization.

Property	Ridge	Lasso
Penalty	L_2: $\sum_j \beta_j^2$	L_1: $\sum_j \lvert\beta_j\rvert$
Shrinks coefficients	Yes	Yes
Exactly zero coefficients	Not usually	Yes, often
Feature selection	Not directly	Built in through sparsity

Key idea

Ridge mainly shrinks coefficients toward zero. Lasso shrinks coefficients and can set some of them exactly to zero, producing a sparse model.

1.3 What does the tuning parameter λ do?

- $\lambda = 0$: the penalty disappears and the objective becomes OLS.
- A larger λ: stronger pressure toward smaller coefficients and often more zeros.
- An excessively large λ: useful predictors may also be removed, causing underfitting.

The best value is usually selected using validation data or cross-validation, not by looking only at training error.

2 The L_1 Penalty and Geometric Intuition

2.1 The L_1 norm

For a coefficient vector $\beta = (\beta_1, \ldots, \beta_p)^T$, its L_1 norm is

$$\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j| \,. \tag{2.1}$$

For example, if $\beta = (2, -3, 4)^T$, then

$$\|\beta\|_1 = |2| + |-3| + |4| = 9. \tag{2.2}$$

Absolute values ensure that positive and negative coefficients do not cancel each other.

2.2 Constraint form and geometry

The penalized problem can be understood through a constrained form:

$$\min_{\beta} \|y - X\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_1 \le t. \tag{2.3}$$

For two coefficients, the L_1 constraint $|\beta_1| + |\beta_2| \le t$ forms a diamond. The Ridge constraint $\beta_1^2 + \beta_2^2 \le t$ forms a circle. The contours of the residual sum of squares are ellipses (under a

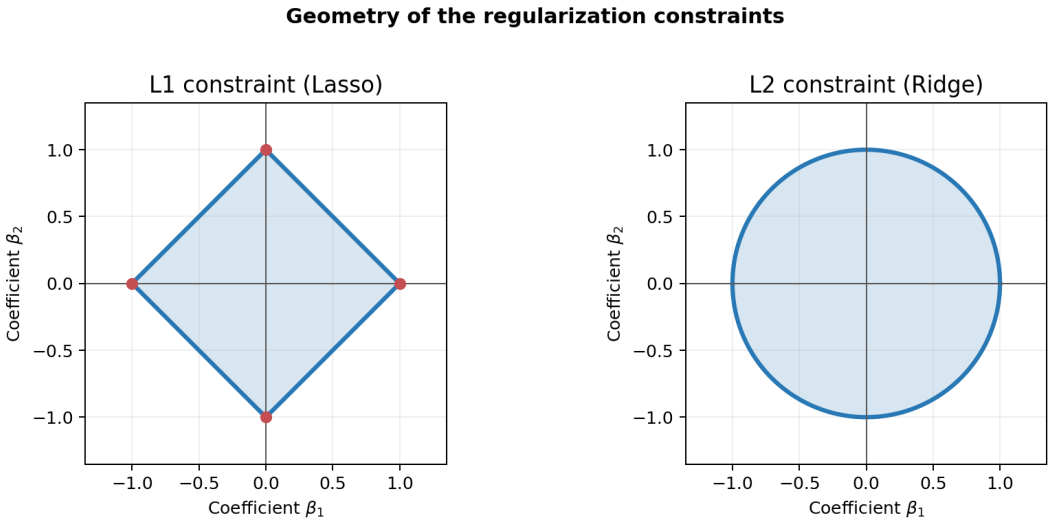

Figure 1: The L_1 constraint has sharp corners on the coordinate axes. The L_2 constraint is smooth.

full-rank design). The optimum occurs where the smallest attainable loss contour touches the constraint region. The corners of the L_1 diamond lie on the axes, where one coefficient is zero. This makes exact zeros more likely for Lasso.

Intuition

The geometry gives a visual explanation for sparsity: the sharp corners of the L_1 region make a solution with one or more zero coefficients more likely. The smooth L_2 boundary does not encourage axis-aligned solutions in the same way.

2.3 Sparsity

A vector is sparse when many of its entries are zero. For example,

$$\hat{\beta} = (2.1, 0, 0, 2.8, 0, -1.1, 0, 0)^T \tag{2.4}$$

has three nonzero coefficients out of eight. The number of nonzero entries is sometimes denoted by the L_0 count:

$$\|\beta\|_0 = \sum_{j=1}^{p} I(\beta_j \neq 0). \tag{2.5}$$

The L_0 count is not a true norm in the strict mathematical sense; it counts nonzero entries. Lasso uses the convex L_1 penalty as a computationally convenient way to encourage sparsity.

3 Optimization: From OLS to Lasso

3.1 Why OLS has a closed-form solution

The OLS objective is

$$J(\beta) = (y - X\beta)^T(y - X\beta) \tag{3.1}$$
$$= y^T y - 2\beta^T X^T y + \beta^T X^T X\beta. \tag{3.2}$$

Differentiating,

$$\nabla_\beta J = -2X^T y + 2X^T X\beta. \tag{3.3}$$

Set the gradient to zero:

$$X^T X\hat{\beta} = X^T y. \tag{3.4}$$

If $X^T X$ is invertible,

$$\boxed{\hat{\beta}_{\text{OLS}} = (X^T X)^{-1} X^T y.} \tag{3.5}$$

3.2 Why Ridge also has a closed form

Ridge adds $\lambda\beta^T\beta$:

$$J_R(\beta) = (y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta. \tag{3.6}$$

Its gradient is

$$-2X^T y + 2X^T X\beta + 2\lambda\beta. \tag{3.7}$$

Setting it to zero gives

$$\boxed{\hat{\beta}_R = (X^T X + \lambda I)^{-1} X^T y.} \tag{3.8}$$

3.3 Why Lasso needs a different optimization method

Lasso adds $\lambda \sum_j |\beta_j|$. The absolute-value function is not differentiable at zero:

$$\frac{d}{dx}|x| = \begin{cases} 1, & x > 0, \\ -1, & x < 0, \\ \text{does not exist}, & x = 0. \end{cases} \tag{3.9}$$

Therefore, we cannot simply differentiate the objective in the ordinary way at every point and solve one linear system. Lasso is still a convex optimization problem and can be solved using subgradients, coordinate descent, proximal-gradient methods, and other algorithms.

3.4 Subgradients

A number g is a subgradient of a convex function f at x if

$$f(z) \geq f(x) + g(z - x) \qquad \text{for every } z. \tag{3.10}$$

For $f(x) = |x|$, the subdifferential is

$$\boxed{\partial |x| = \begin{cases} \{1\}, & x > 0, \\ [-1, 1], & x = 0, \\ \{-1\}, & x < 0. \end{cases}} \tag{3.11}$$

At zero, every slope between -1 and 1 is a valid subgradient. This interval is the mathematical feature that allows the optimum to occur exactly at zero.

For the Lasso objective in **??**, the subgradient optimality condition is

$$\boxed{0 \in 2X^T(X\hat{\beta} - y) + \lambda\, \partial \left\|\hat{\beta}\right\|_1.} \tag{3.12}$$

Because the objective is convex, this condition is necessary and sufficient for a global minimum.

4 KKT Optimality Conditions

4.1 Coefficient-wise KKT conditions

Let x_j denote column j of X, and let $r = y - X\hat{\beta}$ be the residual at the estimated coefficients. The jth coordinate of **??** is

$$0 \in -2x_j^T r + \lambda \partial \left|\hat{\beta}_j\right|. \tag{4.1}$$

Using the three cases in **??**, we obtain:

$$\boxed{\begin{array}{ll} 2x_j^T r = \lambda, & \hat{\beta}_j > 0, \\ 2x_j^T r = -\lambda, & \hat{\beta}_j < 0, \\ \left|2x_j^T r\right| \leq \lambda, & \hat{\beta}_j = 0. \end{array}} \tag{4.2}$$

These are the coordinate-wise KKT conditions for the unnormalized RSS convention used throughout this guide.

Key idea

For a nonzero coefficient, the residual-feature correlation reaches the positive or negative threshold. For a zero coefficient, its absolute value must be no larger than the threshold. This is the KKT explanation for Lasso's exact zeros.

4.2 A small interpretation

Suppose $\lambda = 4$. For a coefficient estimated as zero, the condition is

$$\left|2x_j^T r\right| \leq 4 \quad \Longleftrightarrow \quad \left|x_j^T r\right| \leq 2. \tag{4.3}$$

If the feature's correlation with the current residual is within this range, the zero coefficient is compatible with optimality. If it is outside the range, that coordinate cannot satisfy the zero-coefficient KKT condition at that point.

Important note

Some books and software use $\frac{1}{2}\,\mathrm{RSS} + \lambda \|\beta\|_1$ or $\frac{1}{2n}\,\mathrm{RSS} + \alpha \|\beta\|_1$. Their KKT thresholds have different factors. Always derive the condition from the exact objective convention being used.

5 Soft Thresholding

5.1 The idea

Soft thresholding moves values toward zero. Values whose absolute magnitude is no greater than the threshold become exactly zero. Larger values are reduced in magnitude by the threshold.

$$\boxed{\mathcal{S}(z,t) = \mathrm{sign}(z)\max(|z| - t, 0).} \tag{5.1}$$

Equivalently,

$$\mathcal{S}(z,t) = \begin{cases} z - t, & z > t, \\ 0, & -t \leq z \leq t, \\ z + t, & z < -t. \end{cases} \tag{5.2}$$

5.2 Deriving the threshold in an orthonormal case

Suppose the columns of X are orthonormal, so $X^T X = I$, and define $z = X^T y$. Ignoring terms independent of β, the objective separates into

$$\sum_{j=1}^{p} \left[(\beta_j - z_j)^2 + \lambda\,|\beta_j|\right]. \tag{5.3}$$

For one coordinate, minimize

$$f(\beta) = (\beta - z)^2 + \lambda\,|\beta|\,. \tag{5.4}$$

If $\beta > 0$, setting the derivative to zero gives $2(\beta - z) + \lambda = 0$, hence $\beta = z - \lambda/2$. If $\beta < 0$, it gives $2(\beta - z) - \lambda = 0$, hence $\beta = z + \lambda/2$. At $\beta = 0$, the subgradient condition holds when $|z| \leq \lambda/2$. Combining the cases,

$$\boxed{\hat{\beta}_j = \mathcal{S}(z_j, \lambda/2).} \tag{5.5}$$

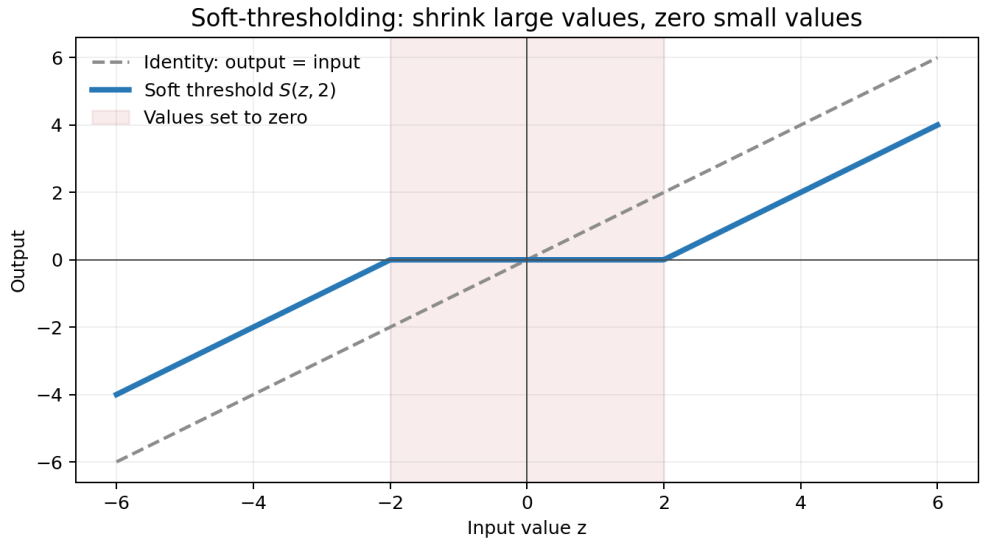

Figure 2: Soft thresholding with threshold $t = 2$. Values in $[-2, 2]$ are set to zero.

Worked example

Let $t = 2$. Then $\mathcal{S}(5, 2) = 3$, $\mathcal{S}(-5, 2) = -3$, and $\mathcal{S}(1.5, 2) = 0$. Large values shrink; small values disappear.

Important note

The threshold is $\lambda/2$ here because the objective uses unnormalized RSS, $\text{RSS} + \lambda \|\beta\|_1$. If the loss is $\frac{1}{2}\,\text{RSS} + \lambda \|\beta\|_1$, the threshold is λ.

6 Coordinate Descent

6.1 What is coordinate descent?

Coordinate descent is an iterative optimization algorithm. Instead of updating all coefficients simultaneously, it updates one coefficient at a time while holding the other coefficients fixed. It cycles through the features until the solution changes very little.

6.2 Deriving the coordinate update

Start with

$$J(\beta) = \|y - X\beta\|_2^2 + \lambda \sum_{k=1}^{p} |\beta_k| \,. \tag{6.1}$$

To update coefficient β_j, hold all other coefficients fixed and define the partial residual

$$\boxed{r_j = y - \sum_{k \neq j} x_k \beta_k.} \tag{6.2}$$

Then $y - X\beta = r_j - x_j\beta_j$. The part of the objective that depends on β_j is

$$f(\beta_j) = \|r_j - x_j\beta_j\|_2^2 + \lambda\,|\beta_j| + \text{constant} \tag{6.3}$$

$$= \beta_j^2 x_j^T x_j - 2\beta_j x_j^T r_j + \lambda\,|\beta_j| + \text{constant}. \tag{6.4}$$

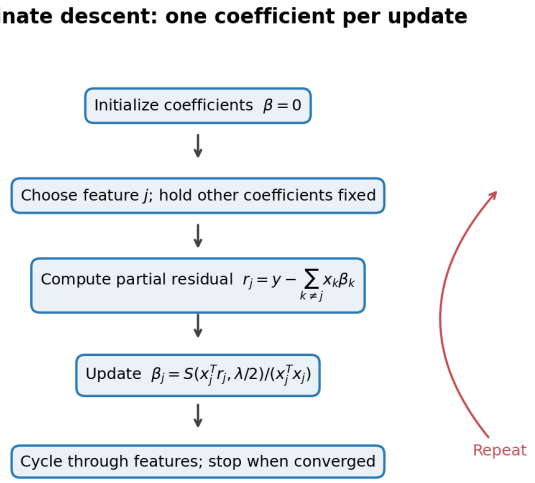

Figure 3: The basic coordinate-descent workflow for Lasso.

Define

$$a_j = x_j^T x_j, \qquad z_j = x_j^T r_j. \tag{6.5}$$

Then

$$f(\beta_j) = a_j \beta_j^2 - 2 z_j \beta_j + \lambda \, |\beta_j| + \text{constant}. \tag{6.6}$$

Completing the square and applying soft thresholding gives

$$\boxed{\beta_j^{\text{new}} = \frac{\mathcal{S}(z_j, \lambda/2)}{a_j} = \frac{\mathcal{S}(x_j^T r_j, \lambda/2)}{x_j^T x_j}.} \tag{6.7}$$

6.3 Numerical coordinate update

Suppose $x_j^T r_j = 7$, $x_j^T x_j = 4$, and $\lambda = 2$. Then the threshold is $\lambda/2 = 1$:

$$\mathcal{S}(7, 1) = 6, \tag{6.8}$$

$$\beta_j^{\text{new}} = \frac{6}{4} = 1.5. \tag{6.9}$$

If instead $x_j^T r_j = 0.5$, then $\mathcal{S}(0.5, 1) = 0$, so the updated coefficient is exactly zero.

6.4 Algorithm in pseudocode

```
Initialize beta = (0, ..., 0)
Repeat until convergence:
  For j = 1, ..., p:
    r = y - X beta + x_j beta_j
    z = x_jT r
    a = x_jT x_j
    beta_j = SoftThreshold(z, lambda/2) / a
```

The residual expression $r = y - X\beta + x_j\beta_j$ removes the current contribution of feature j from the fitted values, leaving the partial residual for that coordinate. The algorithm then updates β_j and moves to the next feature.

6.5 When does the algorithm stop?

A practical stopping rule is

$$\max_j \left|\beta_j^{(t+1)} - \beta_j^{(t)}\right| < \epsilon, \tag{6.10}$$

where ϵ is a small tolerance. Implementations may also monitor objective changes or KKT violations. Since the Lasso objective is convex, coordinate descent converges to a global minimizer under standard conditions; if the minimizer is not unique, different coefficient vectors can attain the same minimum.

6.6 How coordinate descent and KKT connect

Coordinate descent is a method for finding coefficients. KKT conditions characterize whether the coefficients are optimal. At convergence, each coordinate should satisfy the corresponding condition in **??**. Practical solvers check these conditions numerically to assess convergence.

7 Practical Modeling: Scaling and Choosing λ

7.1 Why standardize predictors?

Lasso penalizes coefficient magnitudes. Coefficient magnitudes depend on the units in which predictors are measured. For example, income measured in dollars has a different numerical scale from age measured in years. Without scaling, the penalty can affect features unevenly simply because of their units.

A common standardization is

$$z_{ij} = \frac{x_{ij} - \mu_j}{s_j}, \tag{7.1}$$

where μ_j and s_j are the training-set mean and standard deviation of feature j. After this transformation, each predictor has approximately zero mean and unit standard deviation.

Key idea

Fit the scaler on the training data only. Apply the same training means and standard deviations to validation and test data. This avoids data leakage.

The intercept is usually not penalized. A common approach is to center the predictors and response, fit the penalized slopes, and estimate the intercept separately.

7.2 Choosing lambda with cross-validation

A larger λ often produces a simpler model, but too much regularization can underfit. A common approach is K-fold cross-validation. Split the training data into K folds; train on $K-1$ folds and validate on the remaining fold; repeat so each fold is used for validation; average the validation errors for each candidate λ. Choose the λ with the best average validation performance, or use a simplicity-oriented rule such as the one-standard-error rule. The final test set should be reserved for a final evaluation after model selection.

7.3 Regularization paths

A regularization path shows how each estimated coefficient changes as λ changes. The horizontal axis is often λ on a logarithmic scale, and each curve represents one coefficient. As regularization

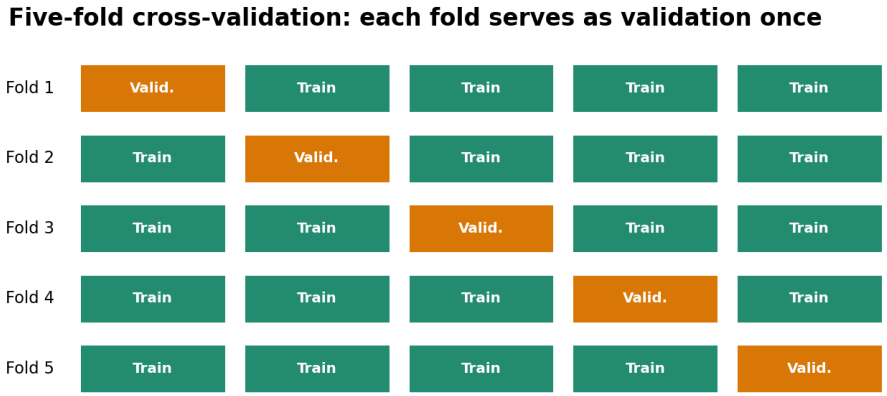

Figure 4: Five-fold cross-validation. Each fold is used as the validation fold once.

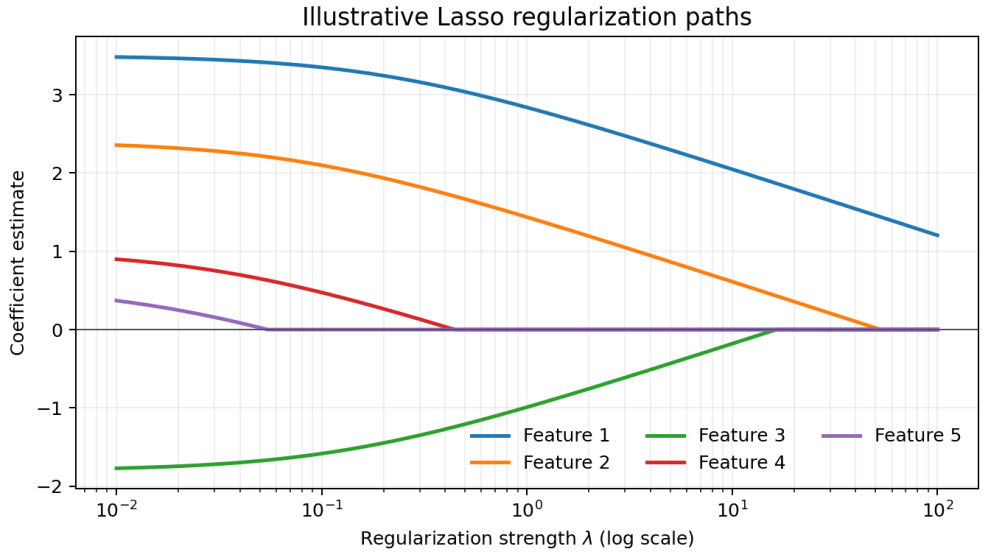

Figure 5: Illustrative coefficient paths as regularization becomes stronger. This is a conceptual illustration, not empirical data.

becomes stronger, coefficients generally move toward zero. A path can reveal which variables remain active over a range of penalty strengths.

7.4 The all-zero threshold

Assume predictors and response are centered and use the unnormalized objective in **??**. At $\beta = 0$, the KKT condition for every coefficient is

$$\left|2x_j^T y\right| \leq \lambda. \tag{7.2}$$

Thus a threshold at which the all-zero coefficient vector is optimal is

$$\boxed{\lambda_{\max} = 2 \max_j \left|x_j^T y\right|.} \tag{7.3}$$

This formula changes if the loss is scaled differently. If an intercept is present, centering is one way to handle it separately.

8 Advanced Concepts: Strengths and Limitations

8.1 Sparsity and feature selection

Lasso can set some estimated coefficients exactly to zero. This provides a form of feature selection and can make a model easier to interpret. However, a zero coefficient means that the fitted model did not retain that feature under the chosen data, penalty, and other predictors. It does not prove that the feature is scientifically irrelevant or has no causal effect.

8.2 Correlated predictors

Suppose two predictors contain almost the same information. Lasso may retain one and set the other to zero. With perfectly identical columns, multiple coefficient allocations can produce the same predictions and the same L_1 penalty. With highly correlated but not identical columns, the selected feature can be sensitive to small changes in the data.

Ridge often distributes coefficient weight across correlated predictors because the squared L_2 penalty can be smaller when weight is spread across them. Elastic Net combines both penalties:

$$J_{\text{EN}}(\beta) = \text{RSS}(\beta) + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2. \tag{8.1}$$

The L_1 term encourages sparsity; the L_2 term encourages more stable, shared shrinkage among correlated predictors.

8.3 Limitations of Lasso

1. **Unstable selection among correlated predictors.** Small changes in the sample can change which member of a correlated group is selected.
2. **Regularization bias.** Shrinking coefficients toward zero introduces bias, even for predictors that truly matter.
3. **Over-regularization.** A very large λ can remove useful predictors and underfit.
4. **Scaling sensitivity.** Predictors should generally be standardized so that units do not dominate the penalty.
5. **No causal guarantee.** Predictive feature selection does not establish causation.

6. **Dense true models.** If many predictors genuinely have small nonzero effects, Lasso may remove some of them; Ridge may be more appropriate.
7. **High-dimensional selection limit.** Under common general-position conditions, a Lasso solution with n observations selects no more than n predictors. This can matter when the number of predictors greatly exceeds the sample size.

9 Revision Sheet and Self-Test

9.1 The essential formulas

$$\begin{aligned}
\text{Lasso objective:} &\quad \text{RSS} + \lambda \|\beta\|_1 \\
L_1 \text{ norm:} &\quad \|\beta\|_1 = \sum_j |\beta_j| \\
\text{Absolute-value subgradient:} &\quad \partial |x| = \begin{cases} \{1\}, & x > 0 \\ [-1, 1], & x = 0 \\ \{-1\}, & x < 0 \end{cases} \\
\text{Lasso optimality:} &\quad 0 \in 2X^T(X\hat{\beta} - y) + \lambda \partial \left\|\hat{\beta}\right\|_1 \\
\text{Soft threshold:} &\quad \mathcal{S}(z, t) = \text{sign}(z) \max(|z| - t, 0) \\
\text{Coordinate update:} &\quad \beta_j^{\text{new}} = \frac{\mathcal{S}(x_j^T r_j, \lambda/2)}{x_j^T x_j}.
\end{aligned}$$

9.2 Conceptual summary

- Lasso is linear regression with an L_1 penalty.
- The L_1 penalty encourages sparse coefficient vectors.
- The absolute value is not differentiable at zero, so subgradients are used.
- KKT conditions characterize the optimal coefficient pattern.
- Soft thresholding shrinks large values and sets sufficiently small values to zero.
- Coordinate descent updates one coefficient at a time using partial residuals.
- Standardization and cross-validation are important for practical modeling.
- Correlated predictors and regularization bias are important limitations.

9.3 Self-test questions

1. What is the difference between Ridge and Lasso penalties?
2. Why can Lasso produce exact zeros?
3. Why is the ordinary derivative of $|x|$ unavailable at $x = 0$?
4. State the three coefficient-wise KKT cases for Lasso.
5. What does soft thresholding do when $|z| \leq t$?
6. Why does coordinate descent use a partial residual?
7. Derive the coordinate update formula from the one-coordinate objective.

8. Why should predictors generally be standardized before fitting Lasso?
9. How does cross-validation help select λ?
10. Why can highly correlated predictors make Lasso feature selection unstable?

9.4 Short answers

1. Ridge uses an L_2 squared penalty; Lasso uses an L_1 absolute-value penalty.
2. The L_1 penalty has a subgradient interval at zero, and its geometry has corners on the coordinate axes.
3. The left and right derivatives at zero differ (-1 and $+1$).
4. Positive: $2x_j^T r = \lambda$; negative: $2x_j^T r = -\lambda$; zero: $\left|2x_j^T r\right| \leq \lambda$.
5. It returns zero.
6. The partial residual removes the current contribution of feature j, isolating the one-dimensional update.
7. Expand the one-coordinate objective, complete the square, and apply soft thresholding.
8. The penalty acts on coefficient magnitudes, which depend on predictor units.
9. It estimates out-of-sample performance for candidate penalty values.
10. Similar predictors can substitute for one another, so one may be selected while another is discarded.

One final mental model

Lasso balances fit and simplicity. The residual term asks the model to explain the data; the L_1 term asks it to keep coefficient magnitudes under control. Subgradients and KKT conditions describe the optimum, soft thresholding explains zero coefficients, and coordinate descent is a practical way to find them.