

# 回归的正则化方法：岭回归、Lasso 与 Elastic Net

2025 年 9 月 6 日

## 目录

### 1 引言

正则化用于控制模型复杂度，以缓解过拟合并提升泛化。在线性回归中，常见方法包括岭回归 ( $\ell_2$ )、Lasso ( $\ell_1$ ) 与 Elastic Net (两者的组合)。它们通过将系数向零收缩来降低方差；其中 Lasso 还能产生稀疏性，从而实现嵌入式特征选择。

### 2 原理与公式

#### 2.1 模型

给定特征  $\mathbf{x} \in \mathbb{R}^d$ ，预测  $\hat{y} = \mathbf{w}^\top \mathbf{x} + b$ 。通常不对截距  $b$  加正则。

#### 2.2 岭回归 ( $\ell_2$ )

$$\min_{\mathbf{w}, b} \frac{1}{2n} \|\mathbf{X}\mathbf{w} + b\mathbf{1} - \mathbf{y}\|_2^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2. \quad (1)$$

在不惩罚截距时，可写闭式解  $\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top (\mathbf{y} - \bar{y}\mathbf{1})$ 。岭回归连续收缩系数、缓解多重共线性，但不产生严格的零系数。

#### 2.3 Lasso ( $\ell_1$ )

$$\min_{\mathbf{w}, b} \frac{1}{2n} \|\mathbf{X}\mathbf{w} + b\mathbf{1} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_1. \quad (2)$$

鼓励稀疏（部分  $w_j = 0$ ）。由次梯度/KKT 条件可知：当残差与特征的相关性落入  $[-\lambda, \lambda]$  带内时，对应系数会被压为零。

## 2.4 Elastic Net

结合  $\ell_1$  与  $\ell_2$ ：

$$\min_{\mathbf{w}, b} \frac{1}{2n} \|\mathbf{X}\mathbf{w} + b\mathbf{1} - \mathbf{y}\|_2^2 + \lambda \left( \alpha \|\mathbf{w}\|_1 + \frac{1-\alpha}{2} \|\mathbf{w}\|_2^2 \right), \alpha \in [0, 1]. \quad (3)$$

在强相关特征场景下更稳健： $\ell_2$  部分提升稳定性， $\ell_1$  保持稀疏。

## 2.5 标准化与截距

建议对特征逐列标准化、对  $\mathbf{y}$  去中心化；截距  $b$  不加正则并在居中数据上单独估计。

## 2.6 优化

岭回归可用闭式解或 QR/SVD；Lasso 与 Elastic Net 常用坐标下降与软阈值（配合热启动与递减  $\lambda$  路径）。

# 3 应用场景与要点

- 多重共线性：优先考虑岭回归或 Elastic Net 提升稳定性；
- 特征选择与可解释性：Lasso/Elastic Net 能产生稀疏解；
- 模型选择：通过交叉验证选择  $\lambda/\alpha$ ，结合系数路径与验证曲线判断；
- 预处理：标准化、异常值处理、根据业务对特征进行分组与筛选。

## 4 Python 实战：Ridge 与 Lasso 系数路径

本示例生成合成数据，并分别绘制 Lasso 与岭回归的系数路径，保存为 `figures/lasso_path.png` 与 `figures/ridge_path.png`。

Listing 1: `gen_regularization_figures.py`

```
1 import os
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from sklearn.linear_model import lasso_path, Ridge
5
```

```
6 np.random.seed(7)
7
8 n, d = 120, 12
9 X_raw = np.random.randn(n, d)
10 X_raw[:, 1] = 0.7*X_raw[:, 0] + 0.3*np.random.randn(n)
11
12 true_w = np.zeros(d)
13 true_w[[0, 3, 7]] = [2.0, -3.0, 1.5]
14 y = X_raw @ true_w + 0.8*np.random.randn(n)
15
16 # 标准化特征、去中心化 y
17 X = (X_raw - X_raw.mean(axis=0)) / X_raw.std(axis=0)
18 y = y - y.mean()
19
20 # Lasso 路径 (alpha 递减)
21 alphas_lasso, coefs_lasso, _ = lasso_path(X, y, alphas=None)
22
23 fig, ax = plt.subplots(figsize=(7, 4.5))
24 for j in range(d):
25     ax.plot(alphas_lasso, coefs_lasso[j, :], lw=1.6, label=f"w{j}")
26 ax.set_xscale('log'); ax.invert_xaxis()
27 ax.set_xlabel('alpha (log)'); ax.set_ylabel('coefficient')
28 ax.set_title('Lasso coefficient paths')
29 handles, labels = ax.get_legend_handles_labels()
30 ax.legend(handles[:6], labels[:6], loc='best', fontsize=8)
31
32 fig_dir = os.path.join(
33     "0_Machine Learning", "0_Supervised Learning", "1_Regularization
34     Methods in Regression", "figures")
35 os.makedirs(fig_dir, exist_ok=True)
36 plt.tight_layout(); plt.savefig(os.path.join(fig_dir, 'lasso_path.png'),
37     , dpi=160)
38
39 # 岭回归路径: 不同 alpha 的系数
40 alphas_ridge = np.logspace(-3, 2, 40)
41 coefs_ridge = []
42 for a in alphas_ridge:
43     model = Ridge(alpha=a, fit_intercept=False)
44     model.fit(X, y)
45     coefs_ridge.append(model.coef_)
46 coefs_ridge = np.array(coefs_ridge)
47
48 fig, ax = plt.subplots(figsize=(7, 4.5))
```

```

47 for j in range(d):
48     ax.plot(alphas_ridge, coefs_ridge[:, j], lw=1.6, label=f"w{j}")
49 ax.set_xscale('log'); ax.invert_xaxis()
50 ax.set_xlabel('alpha (log)'); ax.set_ylabel('coefficient')
51 ax.set_title('Ridge coefficient paths')
52 handles, labels = ax.get_legend_handles_labels()
53 ax.legend(handles[:6], labels[:6], loc='best', fontsize=8)
54
55 plt.tight_layout(); plt.savefig(os.path.join(fig_dir, 'ridge_path.png'),
    , dpi=160)
56 print('saved to', os.path.join(fig_dir, 'lasso_path.png'), 'and
    ridge_path.png')

```

## 5 运行效果

图 ?? 与图 ?? 展示了系数随正则强度变化的路径：Lasso 产生稀疏；岭回归连续收缩但不为零。

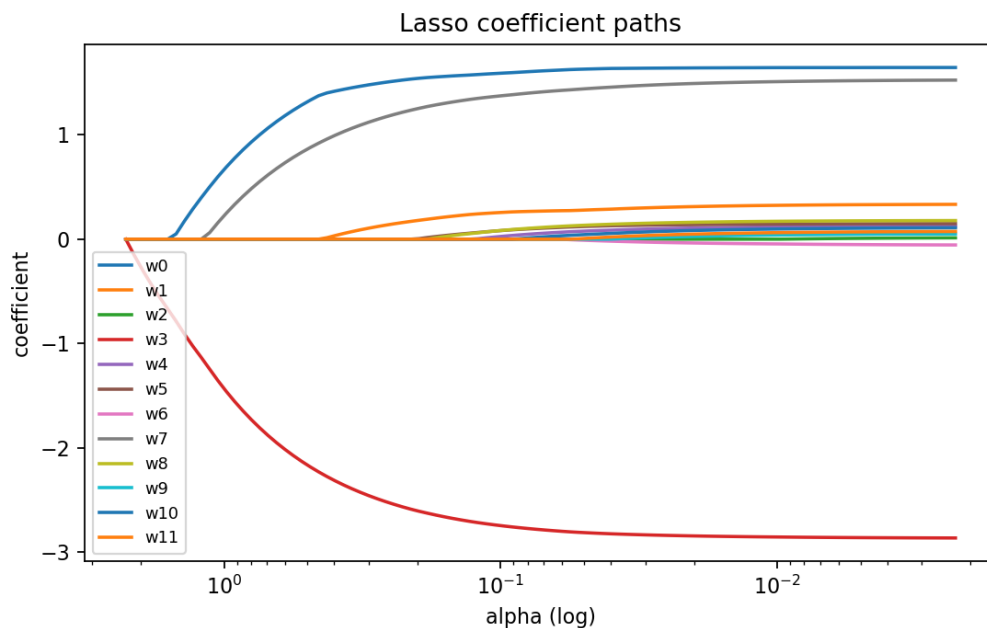


图 1: Lasso 系数路径（合成数据）

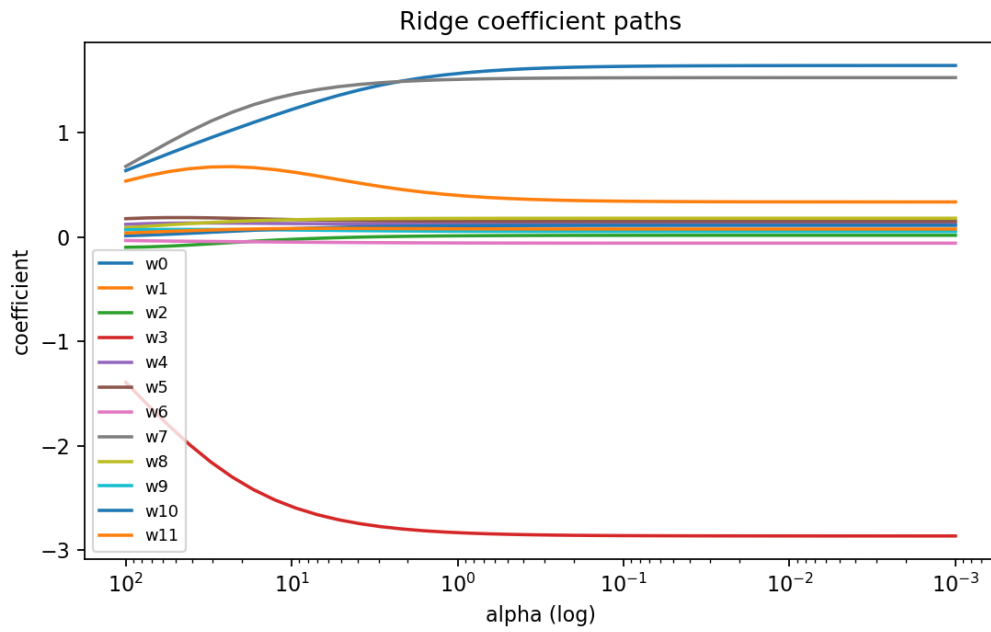


图 2: 岭回归系数路径 (合成数据)

## 6 小结

正则化通过收缩系数来控制方差并提升泛化：多重共线性偏好岭回归，追求稀疏与可解释性使用 Lasso，强相关特征时考虑 Elastic Net。务必进行标准化，并通过交叉验证选择超参数。