

```

In [1]: # 导入操作系统库
import os
# 更改工作目录
os.chdir(r"D:\softwares\applied statistics\pythoncodelearning\chap1\sourcecode")
# 导入基础计算库
import numpy as np
# 导入绘图库
import matplotlib.pyplot as plt
# 导入线性回归模型
from sklearn.linear_model import Ridge
# 导入绘图库中的字体管理包
from matplotlib import font_manager
# 实现中文字符正常显示
font = font_manager.FontProperties(fname=r"C:\Windows\Fonts\SimKai.ttf")
# 使用seaborn风格绘图
plt.style.use("seaborn-v0_8")
# 生成Hilbert矩阵作为X
X = 1.0 / (np.arange(1, 11) + np.arange(0, 10)[:, np.newaxis])
y = np.ones(10)
# 设置不同的惩罚系数alpha的个数
n_alphas = 200
# alpha系数向量
alphas = np.logspace(-10, -2, n_alphas)
# 用来存储估计系数的列表
coefs = []
for a in alphas:
    # 构造岭回归模型
    ridge = Ridge(alpha=a, fit_intercept=False)
    # 模型拟合
    ridge.fit(X, y)
    # 将估计的系数放到列表中
    coefs.append(ridge.coef_)
# 开始绘图
fig, ax = plt.subplots(figsize=(6,6))
# coefs是一个二维列表，每一列是同一个alpha下同一个变量前的系数估计量
ax.plot(alphas, coefs)
# 最横轴做Log变换
ax.set_xscale("log")
# 翻转X轴
ax.set_xlim(ax.get_xlim()[::-1])
# 设置横纵轴标签
plt.xlabel("惩罚系数alpha的对数值", fontproperties=font, fontsize=12)
plt.ylabel("系数的估计值（路径）", fontproperties=font, fontsize=12)
plt.axis("tight")
plt.show()
fig.savefig("../codeimage/code3.pdf")

```

