In [7]:

```
# 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
# 导入数据分析库
import pandas as pd
# 导入模型评估的工具
# 导入数据集获取工具
from sklearn.datasets import load_diabetes
# 导入标准化处理工具
from sklearn.preprocessing import StandardScaler
# 导入Lasso信息准则估计器
from sklearn.linear_model import LassoLarsIC
# 导入管道操作
from sklearn.pipeline import make_pipeline
# 导入时间库
import time
# 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")
# 导入数据集
X, y = load_diabetes(return_X_y=True, as_frame=True)
# 在原始数据集中加入一些随机特征，增加变量
np.random.seed(42)
# 特征数
n_random_features = 14
# 生成随机的X
X_random = pd.DataFrame(
    np.random.randn(X.shape[0], n_random_features),
    columns=[f"random_{i:02d}" for i in range(n_random_features)],
)
# 合并X
X = pd.concat([X, X_random], axis=1)
# 查看下数据
print(X[X.columns[::3]].head())
# 计时开始
start_time = time.time()
# 建立lassoIC模型，它的alpha惩罚系数是自动生成的，无法指定
lasso_lars_aic = make_pipeline(
    StandardScaler(), # 数据标准化
    LassoLarsIC(criterion="aic") # 使用aic准则
)
# 模型拟合
lasso_lars_aic.fit(X, y)
# 记录模型使用的alpha
alpha_aic = lasso_lars_aic[-1].alpha_
# 建立lassoIC模型，它的alpha惩罚系数是自动生成的，无法指定
lasso_lars_bic = make_pipeline(
    StandardScaler(), # 数据标准化
    LassoLarsIC(criterion="bic") # 使用aic准则
)
```

```
# 模型拟合
lasso_lars_bic.fit(X, y)
# 拟合时间
fit_time = time.time() - start_time
print("模型拟合的时间为: ", fit_time, sep="\n")
# 记录模型使用的alpha
alpha_bic = lasso_lars_bic[-1].alpha_
# 将alpha和AIC, BIC存储起来
results = pd.DataFrame(
    {
        "alphas": lasso_lars_aic[-1].alphas_,
        "AIC criterion": lasso_lars_aic[-1].criterion_,
        "BIC criterion": lasso_lars_bic[-1].criterion_
    }
).set_index("alphas")

# 定义一个函数, 选择出最小的AIC对应的alpha
def highlight_min(x):
    x_min = x.min()
    return ["font-weight: bold" if v == x_min else "" for v in x]
# 高亮标记
results.style.apply(highlight_min)
```

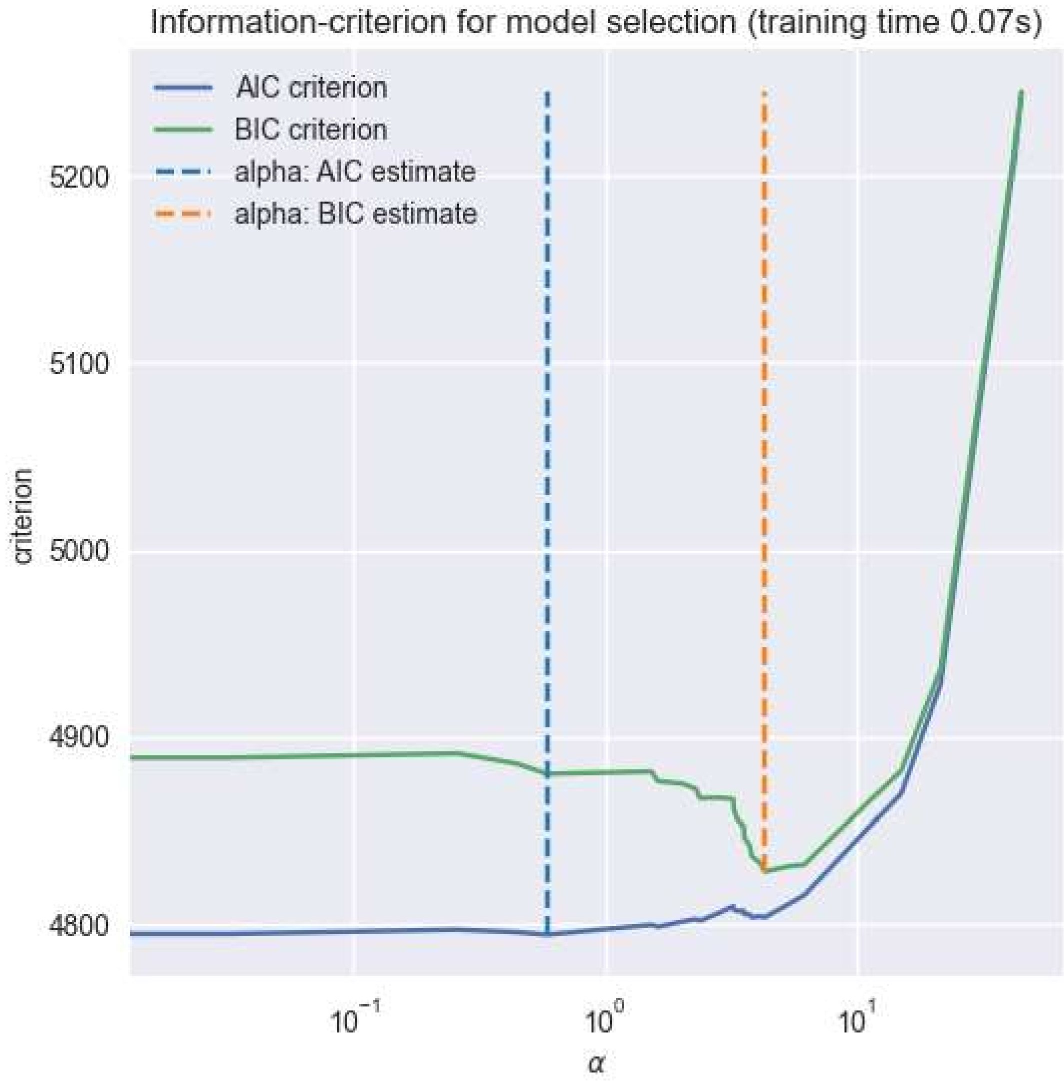
```
        age        bp        s3        s6  random_02  random_05  random_08  \
0  0.038076  0.021872 -0.043401 -0.017646   0.647689  -0.234137  -0.469474
1 -0.001882 -0.026328  0.074412 -0.092204  -1.012831  -1.412304   0.067528
2  0.085299 -0.005670 -0.032356 -0.025930  -0.601707  -1.057711   0.208864
3 -0.089063 -0.036656 -0.036038 -0.009362  -1.478522   1.057122   0.324084
4  0.005383  0.021872  0.008142 -0.046641   0.331263  -0.185659   0.812526

   random_11
0  -0.465730
1   0.110923
2   0.196861
3   0.611676
4   1.003533
模型拟合的时间为:
0.06775593757629395
```

Out[7]:

	AIC criterion	**BIC criterion**
alphas		
45.160030	5244.764779	5244.764779
42.300343	5208.250639	5212.341949
21.542052	4928.018900	4936.201520
15.034077	4869.678359	4881.952289
6.189631	4815.437362	4831.802601
5.329616	4810.423641	4830.880191
4.306012	4803.573491	**4828.121351**
4.124225	4804.126502	4832.765671
3.820705	4803.621645	4836.352124
3.750389	4805.012521	4841.834310
3.570655	4805.290075	4846.203174
3.550213	4807.075887	4852.080295
3.358295	4806.878051	4855.973770
3.259297	4807.706026	4860.893055
3.237703	4809.440409	4866.718747
2.850031	4805.989341	4867.358990
2.384338	4801.702266	4867.163224
2.296575	4802.594754	4872.147022
2.031555	4801.236720	4874.880298
1.618263	4798.484109	4876.218997
1.526599	4799.543841	4881.370039
0.586798	**4794.238744**	4880.156252
0.445978	4795.589715	4885.598533
0.259031	4796.966981	4891.067109
0.032179	4794.662409	4888.762537
0.019069	4794.652739	4888.752867
0.000000	4796.626286	4894.817724

In [8]:

```
# 最后，我们可以绘制不同alpha值的AIC和BIC值。
# 图中的垂直线对应于为每个标准选择的alpha。所选择的α对应于AIC或BIC准则的最小值。
fig1, ax = plt.subplots(figsize=(6,6))
ax = results.plot(ax=ax)
# 画竖直线
ax.vlines(
    alpha_aic,
    results["AIC criterion"].min(),
    results["AIC criterion"].max(),
    label="alpha: AIC estimate",
    linestyles="--",
    color="tab:blue",
)
ax.vlines(
    alpha_bic,
    results["BIC criterion"].min(),
    results["BIC criterion"].max(),
    label="alpha: BIC estimate",
    linestyle="--",
    color="tab:orange",
)
ax.set_xlabel(r"$\alpha$")
ax.set_ylabel("criterion")
ax.set_xscale("log")
# 展示图例
ax.legend()
ax.set_title(
    f"Information-criterion for model selection (training time {fit_time:.2f}s)"
)
plt.show()
fig1.savefig("../codeimage/code11.pdf")
```


Information-criterion for model selection (training time 0.07s)
AIC criterion
BIC criterion
alpha: AIC estimate
alpha: BIC estimate
criterion
10^{-1}
10^{0}
10^{1}
α