

```

In [2]: # 导入操作系统库
import os
# 更改工作目录
os.chdir(r"D:\softwares\applied statistics\pythoncodelearning\chap9\sourcecode")
# 导入基础计算库
import numpy as np
# 导入绘图库
import matplotlib.pyplot as plt
# 导入保序回归
from sklearn.isotonic import IsotonicRegression
# 导入线性回归
from sklearn.linear_model import LinearRegression
# 导入工具
from sklearn.utils import check_random_state
from matplotlib.collections import LineCollection
# 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")
# 生成数据
n = 100
x = np.arange(n)
rs = check_random_state(0)
y = rs.randint(-50, 50, size=(n,)) + 50.0 * np.log1p(np.arange(n))
# 构建保序回归模型
ir = IsotonicRegression(out_of_bounds="clip")
# 模型拟合和预测
y_ = ir.fit_transform(x, y)
# 构建线性模型
lr = LinearRegression()
# 模型拟合
lr.fit(x[:, np.newaxis], y)
# 绘图的一些初始设置
segments = [[[i, y[i]], [i, y_[i]]] for i in range(n)]
lc = LineCollection(segments, zorder=0)
lc.set_array(np.ones(len(y)))
lc.set_linewidths(np.full(n, 0.5))
# 开始绘图
fig, (ax0, ax1) = plt.subplots(ncols=2, figsize=(12, 6))
# 原始数据的散点图
ax0.plot(x, y, "C0.", markersize=12)
# 保序回归下的拟合值
ax0.plot(x, y_, "C1.-", markersize=12)
# 线性回归下的拟合值
ax0.plot(x, lr.predict(x[:, np.newaxis]), "C2-")
ax0.add_collection(lc)
ax0.legend(("Training data", "Isotonic fit", "Linear fit"), loc="lower right")
ax0.set_title("Isotonic regression fit on noisy data (n=%d)" % n)
# 测试集
x_test = np.linspace(-10, 110, 1000)
# 保序回归模型的预测
ax1.plot(x_test, ir.predict(x_test), "C1-")
# 阈值
ax1.plot(ir.X_thresholds_, ir.y_thresholds_, "C1.", markersize=12)
ax1.set_title("Prediction function (%d thresholds)" % len(ir.X_thresholds_))

```

```
plt.show()  
fig.savefig("../codeimage/code1.pdf")
```

