

ロバストな回帰直線 least sum abs

青木繁伸

2020年3月17日

1 目的

観察値と予測値の差の絶対値の p 乗の和を最小にするという方法による回帰直線のパラメータを推定する。

p が 2 のときは、普通の最小二乗法である。

p が 1 のときは、絶対値の和を最小とするものである。

2 使用法

```
import sys
sys.path.append("statlib")
from multi import least_sum_abs
least_sum_abs(x, y, bootstrap=1, a=1, sig=0.95, verbose=True)
```

2.1 引数

x	変数ベクトル
y	変数ベクトル
bootstrap	2以上の値を指定することにより、ブートストラップ法によるパラメータ（切片と傾き）の信頼限界を求める（少なくとも1000以上が望ましい）
p	1のとき絶対値の和を最小にする（2のときは普通の最小二乗法）
sig	信頼率（デフォルトは0.95）
verbose	必要最小限のプリント出力をする

2.2 戻り値の名前

"intercept"	切片
"slope"	傾き
"interceptCL"	bootstrapを2以上にしたときに、切片の信頼限界
"slopeCL"	bootstrapを2以上にしたときに、傾きの信頼限界

3 使用例

```
x = [2,1,3,2,4,3,5,6,5,4,1,4,5,6]
y = [2,3,2,4,5,4,6,6,7,5,6,6,5,6]
```

```
import sys
sys.path.append("statlib")
from multi import least_sum_abs
```

```
a = least_sum_abs(x, y)
```

Intercept: 2.9992880872288605

Slope: 0.5001198665578978

```
import numpy as np
np.random.seed(123) # 乱数の種 (通常は不要)
b = least_sum_abs(x, y, bootstrap=1000)
```

	Estimate	Lower C.L.	Upper C.L.
Intercept	2.999288	0.000002	5.999978
Slope	0.500120	0.000003	1.194444

```
import numpy as np

def simple_reg(x, y):
    mx = np.mean(x)
    my = np.mean(y)
    slope = sum((x-mx)*(y-my))/sum((x-mx)**2)
    intercept = my-slope*mx
    return intercept, slope
```

```
import matplotlib.pyplot as plt

intercept = a["intercept"]
slope = a["slope"]
x0 = np.amin(x)
x1 = np.amax(x)
x2 = np.array([x0, x1])
y2 = intercept + slope * x2
intercept_reg, slope_reg = simple_reg(x, y)
y3 = intercept_reg + slope_reg * x2
plt.scatter(x, y, c="black", s=9)
plt.plot(x2, y2, label="least_sum_abs", linewidth=0.5, color="red")
plt.plot(x2, y3, label="simple regression", linewidth=0.5, color="
    black")
plt.xlabel("x")
```

```
plt.ylabel("y")
plt.legend()
plt.show()
```

