

シンプレックス法によるパラメータ推定

青木繁伸

2020年3月17日

1 目的

シンプレックス法により 1 変数関数のパラメータを求める。

2 使用法

```
import sys
sys.path.append("statlib")
from misc import simplex
simplex(start, x, y, MAXIT, EPSILON, LO, HI)
```

2.1 引数

<code>start</code>	パラメータの初期値ベクトル
<code>x</code>	x 値ベクトル
<code>y</code>	y 値ベクトル
<code>MAXIT</code>	繰り返し上限数
<code>EPSILON</code>	推定許容誤差
<code>LO</code>	パラメータの初期値ベクトルからパラメータベクトルを作るときの倍数 ^{*1}
<code>HI</code>	パラメータの初期値ベクトルからパラメータベクトルを作るときの倍数

2.2 戻り値の名前

<code>"converged"</code>	収束したなら True
<code>"parameters"</code>	推定されたパラメータ
<code>"residual"</code>	残差 (誤差)

^{*1} LO *leqq* パラメータ *leqq* HI の範囲で新たなパラメータベクトルを作る

3 使用法

x と y において, $y = f(x)$ となる 1 変数関数のパラメータを求める。

$$y = f(x) = \frac{p_0}{1 + p_1 * \exp(-p_2 * x)} \quad (1)$$

```
import numpy as np

def fun2(x, p): # 残差平方が最小値となるパラメータを探す目的関数
    return p[0] / (1 + p[1] * np.exp(-p[2] * x))

x = range(1, 11) # x 値
y = [3, 8, 15, 35, 57, 80, 92, 95, 99, 100] # y 値
start = [80, 70, 0.5]

import sys
sys.path.append("statlib")
from misc import simplex

np.random.seed(123) # 乱数の種の設定 (普通は必要ない)
a = simplex(fun2, start, x, y)
```

```
parameters: [100.17983944 100.93696219 0.98879331]
```

```
residual: 9.999328115233183
```

```
import pandas as pd
import matplotlib.pyplot as plt

pred = fun2(x, a["parameters"])
df = pd.DataFrame({"x": x, "y": y, "pred": pred})
print(df)
```

	x	y	pred
0	1	3	2.598624
1	2	8	6.692056
2	3	15	16.165390
3	4	35	34.150477
4	5	57	58.267650
5	6	80	79.031142
6	7	92	91.109538
7	8	95	96.602034
8	9	99	98.818269
9	10	100	99.668940

```
import numpy as np

x2 = np.arange(0, 11, 0.01)
pred2 = fun2(x2, a["parameters"])
plt.plot(x2, pred2, label="logistic", linewidth=0.5)
plt.xlabel("x")
plt.ylabel("y")
plt.title("curve fit")
plt.scatter(x, pred, c = "red", label="observed")
plt.legend()
```

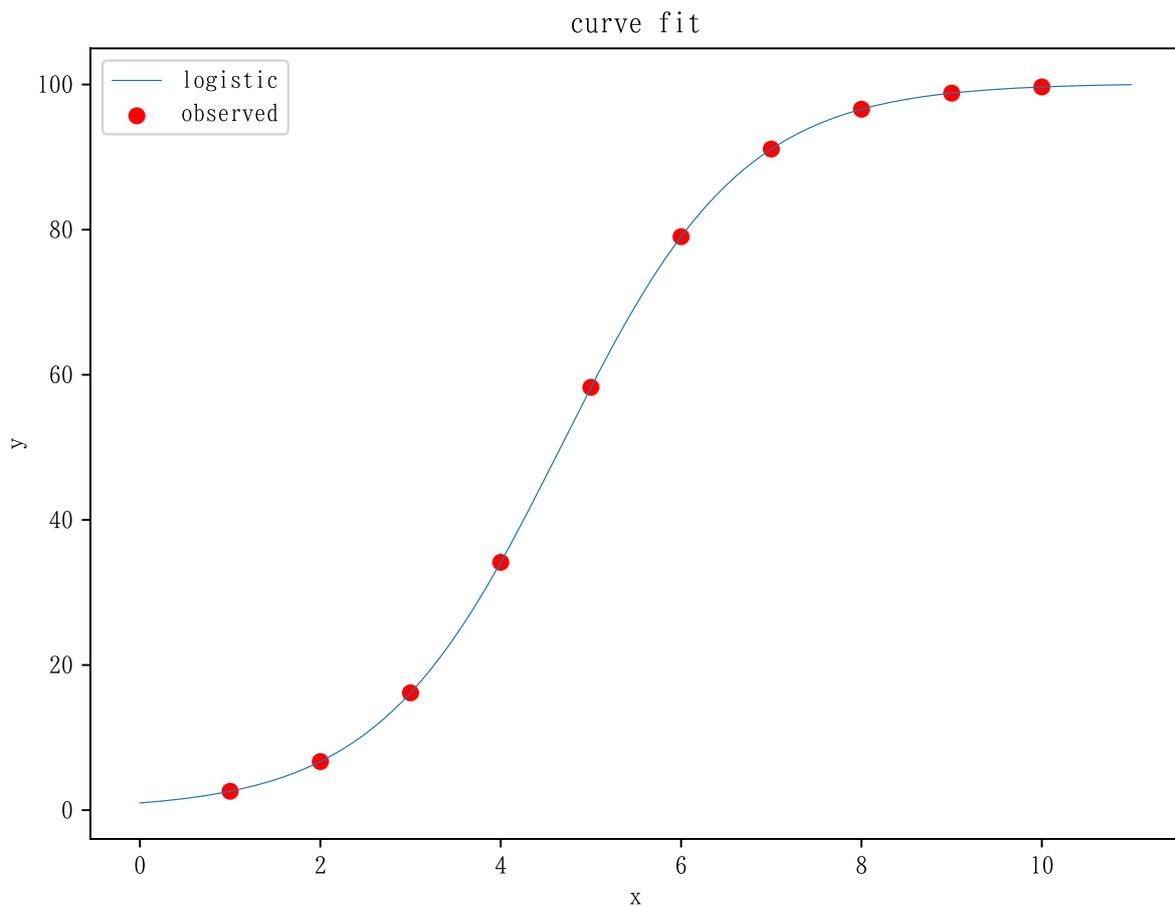


図1 シンプレックス法によるパラメータ推定