線形近似によるロジスティック回帰

青木繁伸

2020 年 3 月 17 日

1　目的

ロジスティック回帰曲線 $y = \dfrac{a}{1 + b\exp(-cx)}$ のパラメータを線形近似により求める。より妥当なあてはめを行う場合には，非線形最小二乗あてはめを行うほうがよい。

独立変数は，1 から始まる連続する整数とする。従属変数は全て正の値でなければならない（0 も不可）。

ロジスティック曲線を表す (1) 式の両辺の逆数をとると，(2) 式のようになる。

$$y = \frac{a}{1 + b\ \exp(-c\ x)} \tag{1}$$

$$\frac{1}{y} = \frac{1}{a} + \frac{b}{a}\exp(-c\ x) \tag{2}$$

ここで，$Y = 1/y$，$A = 1/a$，$B = b/a$ とおくと (3) 式のようになる。

$$Y = A + B\ \exp(-c\ x) \tag{3}$$

(3) 式は，未知のパラメータ A，B については線形であるが，c については非線形である。そこで，以下のような逐次的に近似する手法をとる。

パラメータ c の初期近似値を c_1 とする（$c = c_1 + \delta$）。

$$\exp(-\ x) \fallingdotseq \exp(-c_1\ x) - \delta\ x\ \exp(-c_1\ x) \tag{4}$$

(3) 式に代入して，

$$Y = A + B\{\exp(-c_1\ x) - \delta\ x\ \exp(-c_1\ x)\} \tag{5}$$

$X_1 = \exp(-c_1 x)$，$X_2 = x\ \exp(-c_1 x)$，$C = B\,\delta$ とおくと，

$$Y = A + B\,X_1 - C\,X_2 \tag{6}$$

(6) 式は，2 個の独立変数（X_1，X_2）からなる重回帰式なので，A，B，C を求めることができる。

c の近似値 c_1 の改良値 c_2 は，$\delta = C/B$ であるから，$c_2 = c_1 + \delta$ と表される。$a = 1/A$，$b = a \times B$ である。

パラメータ c の修正量 δ が十分小さくなるまで (6) 式の重回帰式を繰返して計算する。

2　使用法

```
import sys
sys.path.append("statlib")
```

```
from multi import logistic
```

2.1 引数

y	従属変数
maxit	収束計算上限回数（デフォルトで 1000）
tol	反復収束条件（デフォルトで 0.000001）
verbose	必要最小限のプリント出力をする（デフォルトは True）。

2.2 戻り値の名前

"a"	a
"b"	b
"c"	c
"predicted"	予測値
"resid"	残差（観察値 – 予測値）
"x"	独立変数 x
"y"	従属変数 y
"method"	分析手法名

3 使用例

```
import numpy as np

import matplotlib.pyplot as plt

def graph(y, a, b, c):
    x = np.arange(len(y)) + 1
    x0 = np.min(x)
    x1 = np.max(x)
    delta = (x1-x0)*0.05
    x2 = np.arange(x0-delta, x1+delta, (x1-x0+2*delta)/500)
    y2 = a / (1 + b*np.exp(-c*x2))
    plt.scatter(x, y, c="black", s=9)
    plt.plot(x2, y2, linewidth=0.5, color="red")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.show()
```

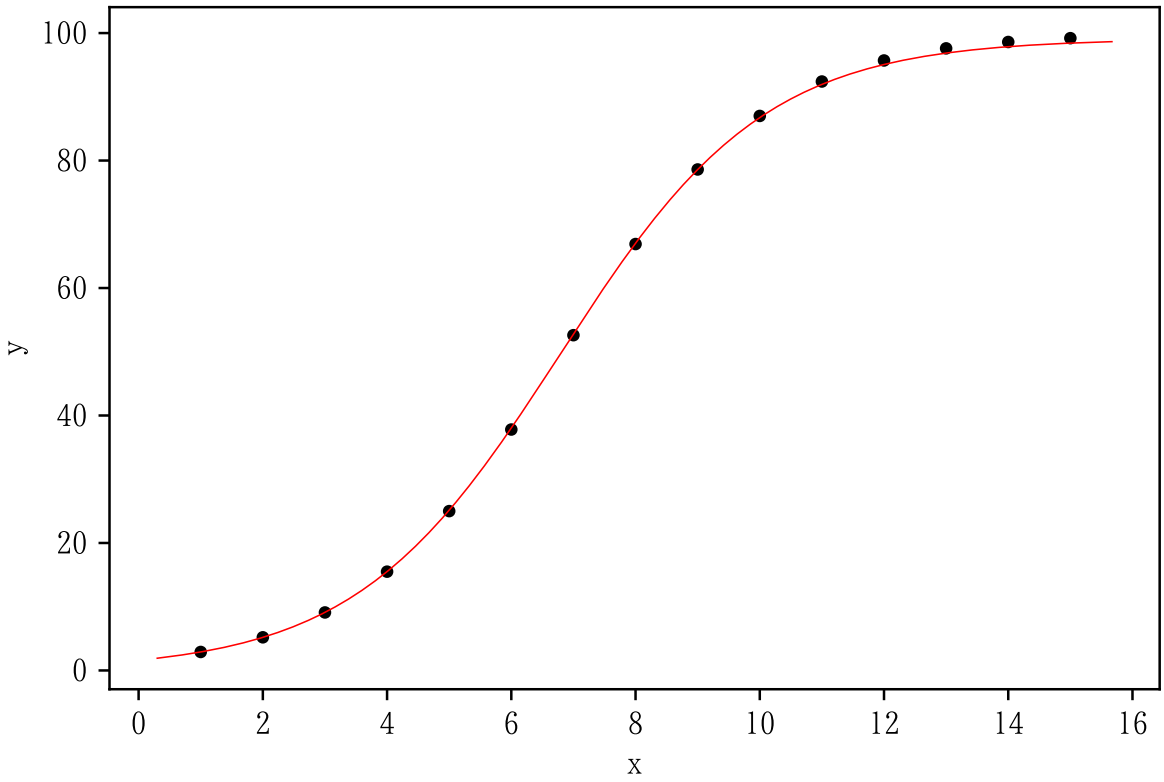
```
import sys
sys.path.append("statlib")
from multi import logistic
```

```
y = [2.9, 5.2, 9.1, 15.5, 25.0, 37.8, 52.6, 66.9, 78.6, 87.0, 92.4,
    95.7, 97.6, 98.6, 99.2]
ans = logistic(y)
```

```
y = a / (1 + b * exp(-c * x))
a = 99.1062, b = 60.7564, c = 0.605524
```

	x	y	pred.	resid.
0	1	2.9	2.901223	-0.001223
1	2	5.2	5.189233	0.010767
2	3	9.1	9.110771	-0.010771
3	4	15.5	15.506534	-0.006534
4	5	25.0	25.138002	-0.138002
5	6	37.8	38.030374	-0.230374
6	7	52.6	52.813748	-0.213748
7	8	66.9	67.036298	-0.136298
8	9	78.6	78.586916	0.013084
9	10	87.0	86.744501	0.255499
10	11	92.4	91.954133	0.445867
11	12	95.7	95.070403	0.629597
12	13	97.6	96.862005	0.737995
13	14	98.6	97.868622	0.731378
14	15	99.2	98.426898	0.773102

```
graph(y, ans["a"], ans["b"], ans["c"])
```


100
80
60
40
20
0
y
0
2
4
6
8
10
12
14
16
x