

# 線形近似によるゴンペルツ曲線回帰

青木繁伸

2020年3月17日

## 1 目的

ゴンペルツ曲線  $y = ab^{\exp(-cx)}$  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)。

ゴンペルツ曲線を表す (1) 式の両辺の対数をとると、(2) 式のようになる。

$$y = a b^{\exp(-c x)} \quad (1)$$

$$\log y = \log a + \log b \exp(-c x) \quad (2)$$

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

$$Y = A + B \exp(-c x) \quad (3)$$

(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  $c_1$  とすると ( $c = c_1 + \delta$ )。

$$\exp(-c x) \doteq \exp(-c_1 x) - \delta x \exp(-c_1 x) \quad (4)$$

(3) 式に代入して、

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

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

$$Y = A + B X_1 - C X_2 \quad (6)$$

(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、 $c_2 = c_1 + \delta$  である。

パラメータ  $c$  の修正量  $\delta$  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 gomperts
```

## 2.1 引数

<code>y</code>	従属変数
<code>maxit</code>	収束計算上限回数 (デフォルトで 1000)
<code>tol</code>	反復収束条件 (デフォルトで 0.000001)
<code>verbose</code>	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

<code>"a"</code>	$a$
<code>"b"</code>	$b$
<code>"c"</code>	$c$
<code>"predicted"</code>	予測値
<code>"resid"</code>	残差 (観察値 - 予測値)
<code>"x"</code>	独立変数 $x$
<code>"y"</code>	従属変数 $y$
<code>"method"</code>	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 * 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 gomperts

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 = gomperts(y)
```

```
y = a * b**exp(-c*x)
```

```
a = 134.135, b = 0.00680792, c = 0.224786
```

	x	y	pred.	resid.
0	1	2.9	2.493440	0.406560
1	2	5.2	5.561857	-0.361857
2	3	9.1	10.555993	-1.455993
3	4	15.5	17.609913	-2.109913
4	5	25.0	26.501587	-1.501587
5	6	37.8	36.732508	1.067492
6	7	52.6	47.674639	4.925361
7	8	66.9	58.712182	8.187818
8	9	78.6	69.336534	9.263466
9	10	87.0	79.187098	7.812902
10	11	92.4	88.050641	4.349359
11	12	95.7	95.837286	-0.137286
12	13	97.6	102.548139	-4.948139
13	14	98.6	108.243995	-9.643995
14	15	99.2	113.019613	-13.819613

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

