漸近指数回帰 $y = a * b^x + c$

青木繁伸

2020 年 3 月 17 日

1　目的

漸近指数回帰曲線 $y = a * b^x + c$ への当てはめを行う。

2　使用法

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

2.1　引数

x	独立変数ベクトル（リストでもよい）
y	従属変数ベクトル（リストでもよい）
maxiter	収束計算の回数の上限（デフォルトは 1000）
tol	許容誤差（デフォルトは 1e-6）
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 sys
sys.path.append("statlib")
from multi import asyr
```

```
import matplotlib.pyplot as plt

def graph(x, y, a, b, c):
        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**x2 + c
        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()
```

3.1 使用例 1

```
import numpy as np

x = np.arange(0, 5.5, 0.5)
y = np.array([52, 53, 56, 60, 68, 81, 104, 144, 212, 331, 536])
y = [52, 53, 56, 60, 68, 81, 104, 144, 212, 331, 536]
ans = asyr(x, y)
```

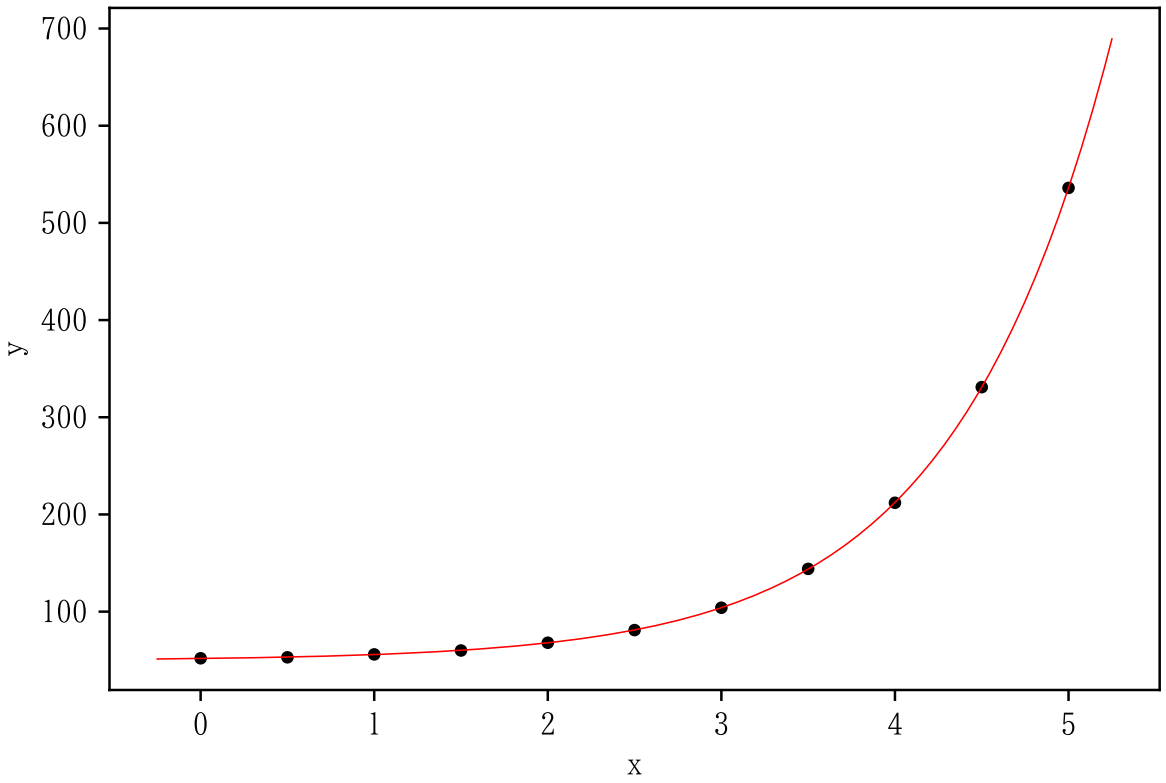
```
y = a * b**x + c
a = 2.02823, b = 2.99196, c = 49.7351

      x    y       pred.    resid.
0   0.0   52   51.763324  0.236676
1   0.5   53   53.243384 -0.243384
2   1.0   56   55.803486  0.196514
3   1.5   60   60.231767 -0.231767
4   2.0   68   67.891489  0.108511
5   2.5   81   81.140729 -0.140729
6   3.0  104  104.058313 -0.058313
7   3.5  144  143.699509  0.300491
8   4.0  212  212.268009 -0.268009
9   4.5  331  330.872886  0.127114
10  5.0  536  536.027103 -0.027103
```

```
print(ans)
```

```
{'a': 2.028234347196685, 'b': 2.991960122729836, 'c': 49.7350895725225, 'predicted': array([ 51
        67.89148927,  81.14072864, 104.05831344, 143.6995093 ,
       212.26800912, 330.87288635, 536.0271035 ]), 'resid': array([ 0.23667608, -0.243384  ,  0
       -0.14072864, -0.05831344,  0.3004907 , -0.26800912,  0.12711365,
       -0.0271035 ]), 'x': array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ]), 'y':
```

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



3.2 使用例 2

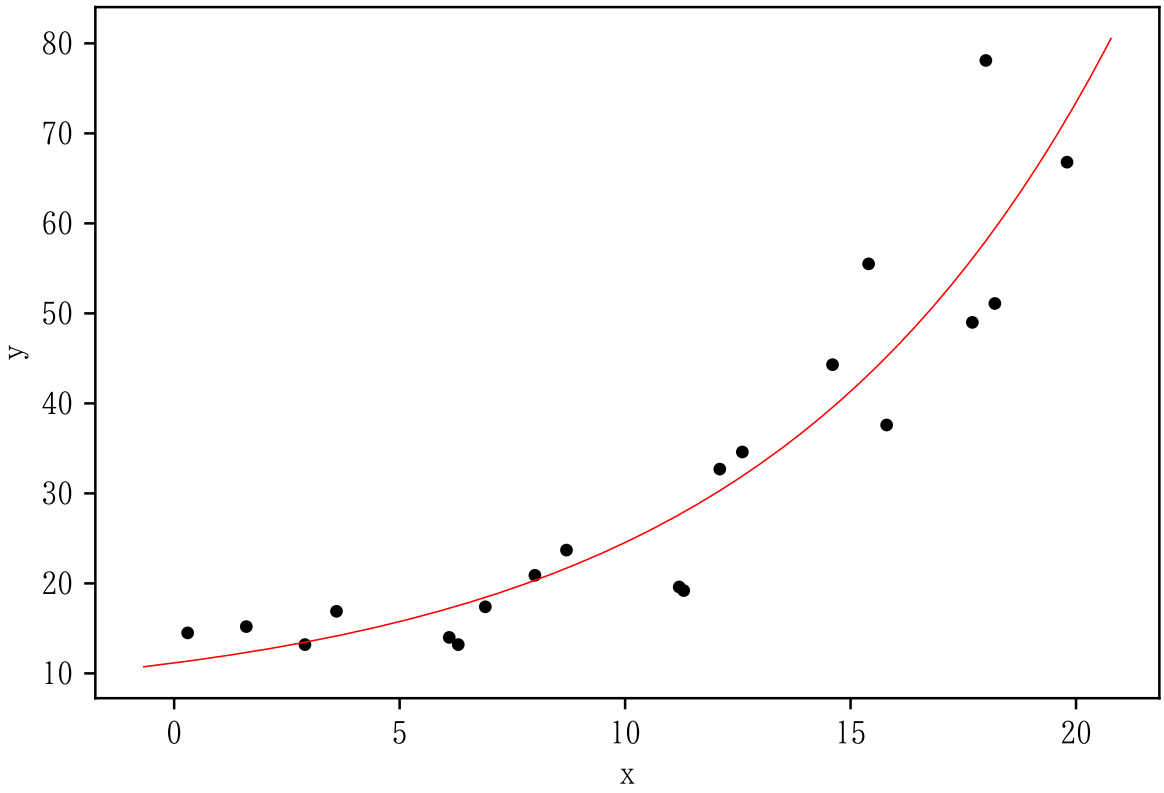
```
x2 = [0.3, 2.9, 3.6, 6.1, 6.9, 8, 11.3, 12.1, 15.8, 15.4, 18, 19.8,
      17.7, 8.7, 1.6, 6.3, 11.2, 12.6, 14.6, 18.2]
y2 = [14.5, 13.2, 16.9, 14, 17.4, 20.9, 19.2, 32.7, 37.6, 55.5, 78.1,
      66.8, 49, 23.7, 15.2, 13.2, 19.6, 34.6, 44.3, 51.1]

ans2 = asyr(x2, y2)
graph(x2, y2, ans2["a"], ans2["b"], ans2["c"])
```

```
y = a * b**x + c
a = 5.04298, b = 1.13834, c = 6.11536

       x     y      pred.     resid.
0    0.3  14.5  11.358220   3.141780
```

1	2.9	13.2	13.458328	-0.258328
2	3.6	16.9	14.155452	2.744548
3	6.1	14.0	17.231074	-3.231074
4	6.9	17.4	18.445097	-1.045097
5	8.0	20.9	20.333782	0.566218
6	11.3	19.2	27.919764	-8.719764
7	12.1	32.7	30.301171	2.398829
8	15.8	37.6	45.178009	-7.578009
9	15.4	55.5	43.205073	12.294927
10	18.0	78.1	58.061934	20.038066
11	19.8	66.8	71.706308	-4.906308
12	17.7	49.0	56.081493	-7.081493
13	8.7	23.7	21.683645	2.016355
14	1.6	15.2	12.320048	2.879952
15	6.3	13.2	17.522886	-4.322886
16	11.2	19.6	27.639071	-8.039071
17	12.6	34.6	31.919888	2.680112
18	14.6	44.3	39.553109	4.746891
19	18.2	51.1	59.425647	-8.325647



非線形回帰の結果はあまり変わらないかもしれない。

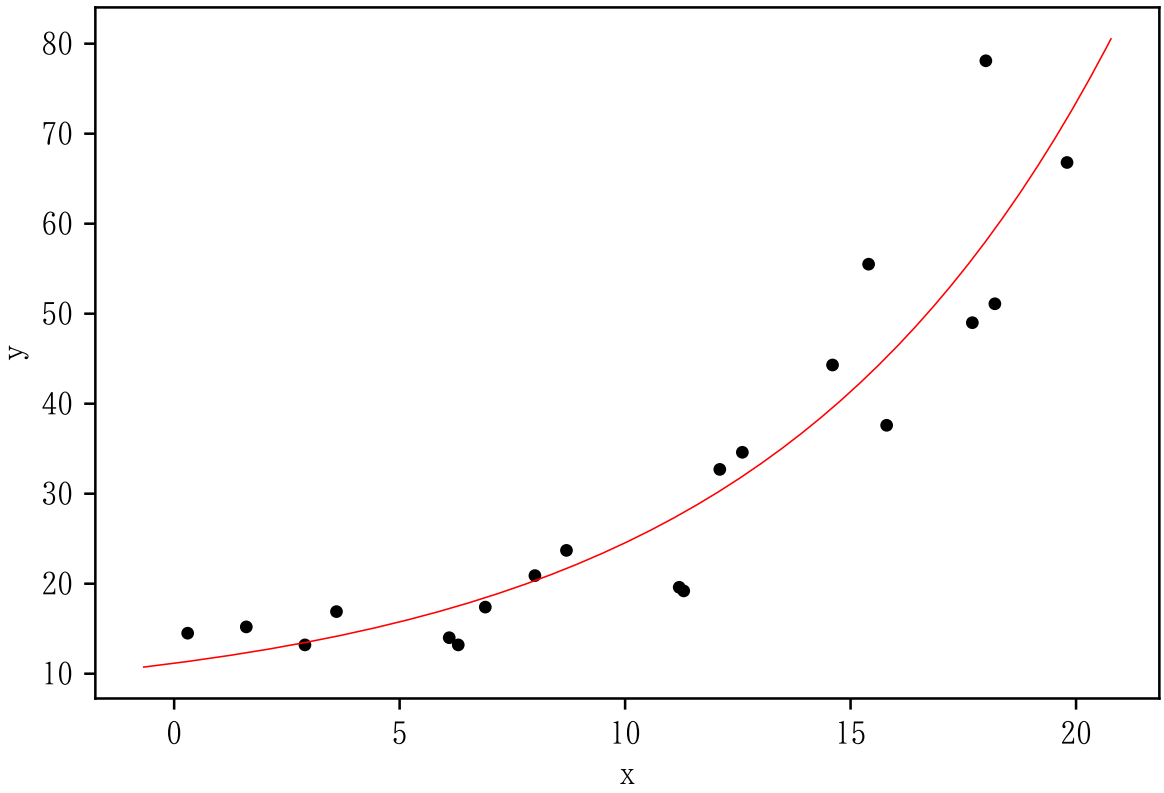
```
from multi import nonlinear_fitting
```

```
ans3 = nonlinear_fitting(x2, y2, [1, 1, 1], model="Asymptotic")
graph(x2, y2, ans3["p"][0], ans3["p"][1], ans3["p"][2])
```

```
Asymptotic      by Marquardt method

                          estimates
a                          5.042981
b                          1.138336
c                          6.115354
residual sum of squares  990.152532

       x     y      pred.     resid.
0    0.3  14.5  11.358217   3.141783
1    2.9  13.2  13.458326  -0.258326
2    3.6  16.9  14.155450   2.744550
3    6.1  14.0  17.231074  -3.231074
4    6.9  17.4  18.445097  -1.045097
5    8.0  20.9  20.333783   0.566217
6   11.3  19.2  27.919766  -8.719766
7   12.1  32.7  30.301173   2.398827
8   15.8  37.6  45.178011  -7.578011
9   15.4  55.5  43.205075  12.294925
10  18.0  78.1  58.061933  20.038067
11  19.8  66.8  71.706303  -4.906303
12  17.7  49.0  56.081493  -7.081493
13   8.7  23.7  21.683647   2.016353
14   1.6  15.2  12.320045   2.879955
15   6.3  13.2  17.522886  -4.322886
16  11.2  19.6  27.639073  -8.039073
17  12.6  34.6  31.919891   2.680109
18  14.6  44.3  39.553111   4.746889
19  18.2  51.1  59.425646  -8.325646
```


80
70
60
50
40
30
20
10
y
0
5
10
15
20
x