関数の最小値を与えるパラメータをシンプレックス法により求める

青木繁伸

1　目的

関数の値が最小となるときのパラメータをシンプレックス法により求める。

2　使用法

```
from simplex2 import simplex2
simplex2(func, start, MAXIT=10000, EPSILON=1e-7, LO=0.8, HI=1.2)
```

func	最小値を求める関数名（ユーザが定義する）
start	関数のパラメータの初期値
MAXIT	繰り返し上限数
EPSILON	推定許容誤差
LO	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]
HI	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

3　戻り値の名前

"converged"	収束したなら True，そうでなければ False
"parameters"	最小値となるときのパラメータ
"minValue"	最小値

4　使用例

4.1　関数の最小値を求める

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2 \tag{1}$$

の最小値を求める。

この関数においては，$x = y = 1$ のときに，最小値 0 となる。

関数を定義する。パラメータは p[0]，p[1]，... に対応させる。

```
def func(p):
    return 100*(p[1]-p[0]**2)**2+(1-p[0])**2
```

しかる後に，パラメータの初期値ベクトルを設定して simplex2() を呼ぶ。

[*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

```
import sys
sys.path.append("statlib")
from simplex2 import simplex2

init = [2.0, 2.0]
simplex2(func, init, EPSILON=1e-12)
```

```
parameters: [1. 1.]
min. value: 8.765365086009888e-26
```

```
{'converged': True,
 'parameters': array([1., 1.]),
 'minValue': 8.765365086009888e-26}
```

4.2 曲線のあてはめを行う

観察値ベクトル x と y があり，複数のパラメータからなる一変数関数 f に対して，y = f(x) となるようパラメータを探索する。

完全なあてはめはできないが，残差平方和 (sum((y-f(x))**2) が最小になるようなパラメータを探索するわけである。

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。

$$y = f(x) = \frac{p_0}{1 + p_1 * \exp(-p_2 * x)} \tag{2}$$

x と y は，大域変数とするので扱いに注意（simplex() の方が簡単）

```
import scipy as sp

def fun(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] * sp.exp(-p[2] * x))

def residual(p): # 残差平方和を求める関数
    return sum((y-fun(p))**2)

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 simplex2 import simplex2

a = simplex2(residual, start, EPSILON=1e-10)
```

```
parameters: [100.17983842 100.93696615   0.98879333]
min. value: 9.99932811522734
```

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

```
x0 = x
pred = fun(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	34.150478
4	5	57	58.267652
5	6	80	79.031143
6	7	92	91.109538
7	8	95	96.602034
8	9	99	98.818268
9	10	100	99.668939

```
import scipy as sp

x = sp.arange(0, 11, 0.01)
pred2 = fun(a["parameters"])
plt.plot(x, pred2, label="logistic", linewidth=0.5)
plt.xlabel("x")
plt.ylabel("y")
plt.title("curve fit")
plt.scatter(x0, pred, c = "red", label="observed")
plt.legend()
```

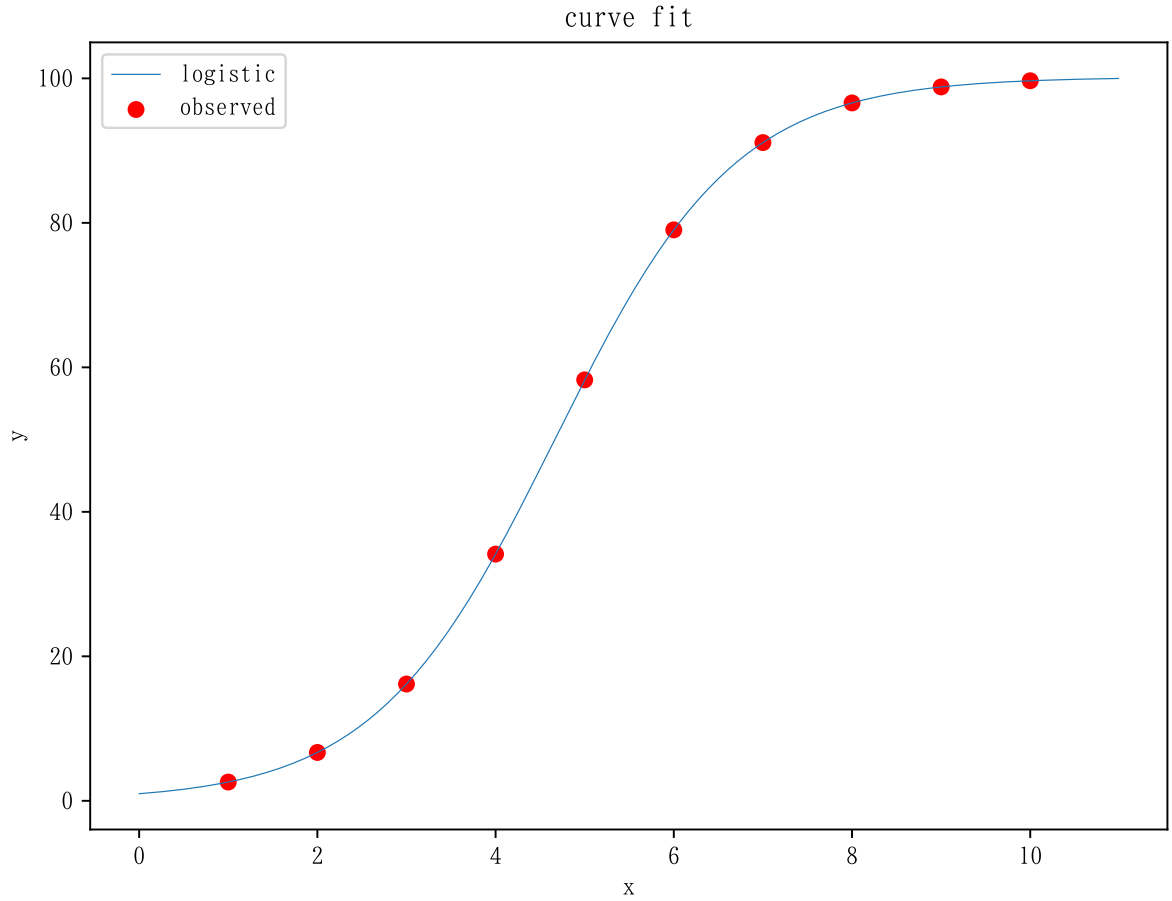

図 1　シンプレックス法によるパラメータ推定