

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

2020年3月17日

1 目的

関数の値が最小となる時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

```
import sys
sys.path.append("statlib")
from misc import simplex2
simplex2(func, start, MAXIT=10000, EPSILON=1e-7, LO=0.8, HI=1.2)
```

2.1 引数

<code>func</code>	最小値を求める関数名（ユーザが定義する）
<code>start</code>	関数のパラメータの初期値
<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 <code>True</code> , そうでなければ <code>False</code>
<code>"parameters"</code>	最小値となる時のパラメータ
<code>"minValue"</code>	最小値

^{*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 使用例

3.1 関数の最小値を求める

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2 \quad (1)$$

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。

```
import sys  
sys.path.append("statlib")  
from misc import simplex2  
  
import numpy as np  
np.random.seed(123) # 乱数の種 (通常は不要)  
  
init = [2.0, 2.0]  
simplex2(func, init, EPSILON=1e-12)
```

```
parameters: [1. 1.]  
min. value: 8.005721000268417e-25  
  
{'converged': True,  
  'parameters': array([1., 1.]),  
  'minValue': 8.005721000268417e-25}
```

3.2 曲線のあてはめを行う

観察値ベクトル \mathbf{x} と \mathbf{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、 $\mathbf{y} = f(\mathbf{x})$ 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 ($\text{sum}((\mathbf{y}-f(\mathbf{x}))^2)$) が最小になるようなパラメータを探索するわけである。

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

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

\mathbf{x} と \mathbf{y} は、大域変数とするので扱いに注意 (`simplex()` の方が簡単)

```
import numpy as np  
  
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] * np.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 misc import simplex2

import numpy as np
np.random.seed(123) # 乱数の種 (通常は不要)

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

```

```

parameters: [100.17983867 100.93695778 0.98879331]
min. value: 9.999328115227987

```

```

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	6.692057
2	3	15	16.165391
3	4	35	34.150479
4	5	57	58.267651
5	6	80	79.031142
6	7	92	91.109538
7	8	95	96.602034
8	9	99	98.818268
9	10	100	99.668939

```

import numpy as np

x = np.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()
```

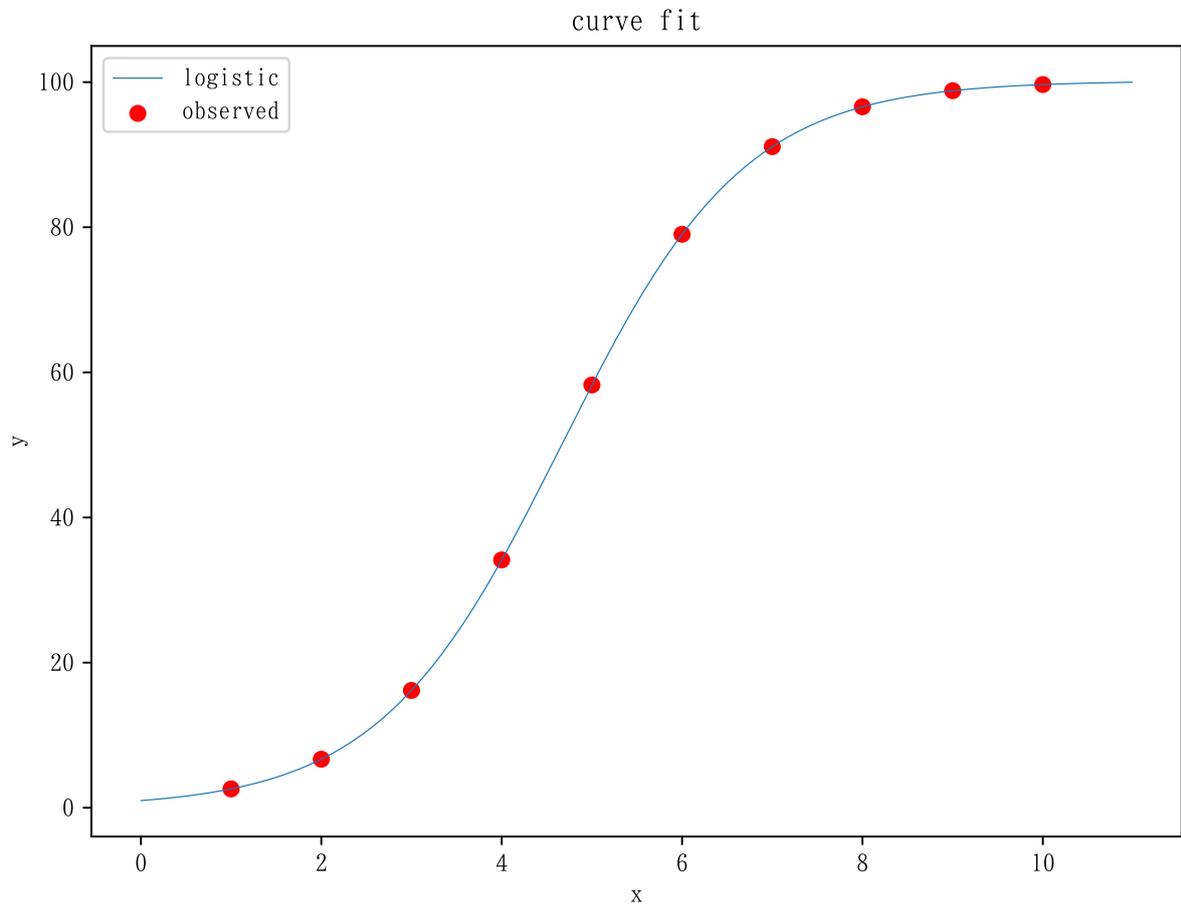


図1 シンプレックス法によるパラメータ推定