



# 最適化の例

金子邦彦



# 最適化

- ある**ゴール**を最小にするように、**パラメータ**を調整すること

**ゴール**： 誤差

**パラメータ**： 直線の上下の位置と、  
直線の傾き

→ 教師データにフィットする  
最適な線分が求まる



*Database Lab.*



ノートページ

# 最適化の例



- 次の式が最小になるように,  $x$  の値を定めなさい。  
但し,  $N=5$  とする。

$$f(\mathbf{x}) = \sum_{i=2}^N 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2.$$

$N=5$  なので,  $x$  は, サイズ5 の 1次元配列である

- <https://docs.scipy.org/doc/scipy/reference/tutorial/optimize.html>

# 最適化を行う Python プログラム



```
import numpy as np

from scipy.optimize import minimize

def rosen(x):
    """The Rosenbrock function"""
    return sum(100.0*(x[1:]-x[:-1])**2.0)**2.0 + (1-x[:-1])**2.0)

x0 = np.array([1.3, 0.7, 0.8, 1.9, 1.2])
res = minimize(rosen, x0, method='nelder-mead',
               options={'xtol': 1e-8, 'disp': True})
print(res.x)
```

# 「最適化を行う Python プログラム」の実行



- Google Colab を使うとき  
次のページの手順 ※ scipy は Google Colab に組み込み済み
- Windows パソコンを使うとき  
前準備として「pip3 install scipy」を実行
- ラズベリーパイ (RaspberryPi) を使うとき  
前準備として「sudo pip3 install scipy」を実行

# Google Colab の利用



Google Colab はオンラインの Python 開発環境.  
するには Google アカウントが必要

① Google Colab のWebページを開く

<https://colab.research.google.com>

② 「ファイル」で、「**PYTHON 3 の新しいノートブックの新規作成**」を選ぶ

③ Google アカウントでログインする

- コードセルの中に Python プログラムを書き、「再生ボタン」をクリック

# 最適化を行う Python プログラム



```
▶ import numpy as np
   from scipy.optimize import minimize
   def rosen(x):
       """The Rosenbrock function"""
       return sum(100.0*(x[1:]-x[:-1])**2.0)**2.0 + (1-x[:-1])**2.0

   x0 = np.array([1.3, 0.7, 0.8, 1.9, 1.2])
   res = minimize(rosen, x0, method='nelder-mead',
                 options={'xtol': 1e-8, 'disp': True})
   print(res.x)
```

```
↳ Optimization terminated successfully.
   Current function value: 0.000000
   Iterations: 339
   Function evaluations: 571
[1. 1. 1. 1. 1.]
```

$x = [1\ 1\ 1\ 1\ 1]$  のとき (すべての値が 1 のとき)  
最適であると求まった。