

使用 scipy 求解带约束的优化问题

- 采用序列二次规划算法

$$\begin{aligned} \min_{x_0, x_1} \quad & 100(x_1 - x_0^2)^2 + (1 - x_0)^2 \\ \text{subject to:} \quad & x_0 + 2x_1 \leq 1 \\ & x_0^2 + x_1 \leq 1 \\ & x_0^2 - x_1 \leq 1 \\ & 2x_0 + x_1 = 1 \\ & 0 \leq x_0 \leq 1 \\ & -0.5 \leq x_1 \leq 2.0. \end{aligned}$$

参考：[Scipy minimize](#)

环境配置

- 为了运行这里的代码，你需要构建一个 Python 环境. 并且安装所需的包：numpy, scipy, matplotlib.
- 这里推荐使用 VScode + Python. 环境配置方法可以参考：<https://zhuanlan.zhihu.com/p/584126712>
- 在上述文章中，执行 `pip install numpy` 时，顺便执行 `pip install scipy` 和 `pip install matplotlib`。
- 环境配置完成后，在 vscode 里打开 `ad_main.ipynb` 并运行就行。

```
In [ ]: import numpy as np
from scipy.optimize import minimize, Bounds
```

目标函数

```
In [ ]: def J(x):
        """目标函数"""
        return 100.0*(x[1]-x[0]**2.0)**2.0 + (1-x[0])**2.0
```

不等式约束

```
In [ ]: ineq_cons = {'type': 'ineq',
                    'fun' : lambda x: np.array([1 - x[0] - 2*x[1],
                                                1 - x[0]**2 - x[1],
                                                1 - x[0]**2 + x[1]])}
```

等式约束

```
In [ ]: eq_cons = {'type': 'eq',
                  'fun' : lambda x: np.array([2*x[0] + x[1] - 1])}
```

优化变量边界

```
In [ ]: bounds = Bounds([0, -0.5], [1.0, 2.0])
```

求解

```
In [ ]: x0 = np.array([0.5, 0]) # 随机猜一个初始解
# Sequential Least Squares Programming (SLSQP) Algorithm (method='SLSQP')
res = minimize(J, x0, method='SLSQP',
              constraints=[eq_cons, ineq_cons], options={'ftol': 1e-9, 'disp': True},
              bounds=bounds)
```

```
Optimization terminated successfully (Exit mode 0)
Current function value: 0.34271757499503896
Iterations: 4
Function evaluations: 13
Gradient evaluations: 4
```

```
In [ ]: print(res.x)
print(res.fun)

[0.41494475 0.1701105 ]
0.34271757499503896
```