最適化手法の実装（2）：Github 公開用

最終更新：2022 年 2 月 10 日

「最適化手法の実装（1）」では，最適化手法の基本的な手法である勾配降下法とそのステップ幅を決めるバックトラック法について簡単に紹介し，Python での実装例を示した．今回は，制約付き最適化問題について，簡単に理論的な話（概要だけ）を紹介してから，実装例を示す．なお，理論的な話は文献 [1] の流れをもとにまとめる．詳細は，後述する参考文献に記した参考書などを参照してほしい．

1 Lagrange の未定乗数法

ここでは一般化した形で議論をすすめていく．次の最適化問題を考える．

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \quad \text{s.t.} \quad g_i(\boldsymbol{x}) = 0, \quad i = 1, \ldots, p \tag{1}$$

なお，変数の次元については $p < n$ を仮定し，関数 $f, g_1, \ldots, g_p$ はすべて微分可能であるとする．

このとき，次の必要条件が成立することが知られている．

1 次の必要条件

局所最適解を $\boldsymbol{x}^* \in \mathbb{R}^n$ とする．また，$\nabla g_1(\boldsymbol{x}^*), \ldots, \nabla g_p(\boldsymbol{x}^*)$ が 1 次独立であるとする．このとき，$\lambda_1^*, \ldots, \lambda_p^* \in \mathbb{R}$ が存在して，

$$\begin{aligned} \nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{p} \lambda_i^* \nabla g_i(\boldsymbol{x}^*) &= \boldsymbol{0} \\ g_i(\boldsymbol{x}) = 0, \quad i = 1, \ldots, p& \end{aligned} \tag{2}$$

が成立する．

式 (2) に出てくる $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p) \in \mathbb{R}^p$ のことを **Lagrange 乗数**といい，式 (2) の $n + p$ 個の方程式を解いて，最適解の候補を絞る手法を **Lagrange の未定乗数法**という．式 (2) の $n + p$ 個の方程式は，

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i=1}^{p} \lambda_i g_i(\boldsymbol{x}) \tag{3}$$

という方程式（**Lagrange 関数**）の $\boldsymbol{x}, \boldsymbol{\lambda}$ での偏微分が 0 になるという式，つまり，

$$\frac{\partial \mathcal{L}}{\partial x_i}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = 0, \quad i = 1, \ldots, n, \qquad \frac{\partial \mathcal{L}}{\partial \lambda_j}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = 0, \quad j = 1, \ldots, p \tag{4}$$

と等価である．Lagrange の未定乗数法は，制約つき最適化問題 (1) の代わりに，制約なし最適化問題 (3) を考える方法ともいえる．Lagrange の未定乗数法ではあくまで最適解の候補しか得られない．その候補が最適解であるかどうかは別の方法で判定する必要がある．

2 双対問題

最適化問題を解くうえで重要な概念の 1 つに**双対性**がある．双対性は，元々の問題を等価な別の問題に置き換えて考える際に利用する性質である．ここで，式 (1) の形の最適化問題のうち，次の線形制約条件をもつ最適化問題を考える．

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{a}_i^\top \boldsymbol{x} - b_i = 0, \quad i = 1, \ldots, p \tag{5}$$

この問題に対して，Lagrange 関数を考えると，

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i=1}^{p} \lambda_i (\boldsymbol{a}_i^\top \boldsymbol{x} - b_i) \tag{6}$$

となる．ここで，

$$\max_{\boldsymbol{\lambda} \in \mathbb{R}^p} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \begin{cases} f(\boldsymbol{x}) & (\boldsymbol{a}_i^\top \boldsymbol{x} - b_i = 0,\ i = 1, \ldots, p) \\ +\infty & (\text{otherwise}) \end{cases} \tag{7}$$

が成立する[1] そのため，式 (5) の問題は

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ \max_{\boldsymbol{\lambda} \in \mathbb{R}^p} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \right\} \tag{8}$$

となる．そして，適当な条件の下では**強双対性**が成立する．

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ \max_{\boldsymbol{\lambda} \in \mathbb{R}^p} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \right\} = \max_{\boldsymbol{\lambda} \in \mathbb{R}^p} \left\{ \min_{\boldsymbol{x} \in \mathbb{R}^n} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \right\} \tag{9}$$

この性質から，$\min f(\boldsymbol{x})$ という本来の最適化問題の代わりに $\phi(\boldsymbol{\lambda}) := \min \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ を注目することが可能である．このとき，$\min f(\boldsymbol{x})$ を**主問題**，$\max \phi(\boldsymbol{\lambda})$ を**双対問題**という．最適化問題の中には，主問題に対応する双対問題を考えると楽になることがある．

3　実装例：双対上昇法の実装

3.1　問題設定

ここでは次の問題を例に双対上昇法の実装例を示す．この問題は高校の数学が知識さえあれば解ける簡単に解ける制約付き最適化問題である．

$$\min_{x, y \in \mathbb{R}} (x^2 + 2y^2) \quad \text{subject to } x + y = 1 \tag{10}$$

まず，高校数学の 2 次関数の話を思い出しながら，この問題の解析解を求めよう．制約条件より $y = 1 - x$ なので，これを式 (10) に代入すると，

$$\min \left\{ x^2 + 2(1-x)^2 \right\} = \min(3x^2 - 4x + 2) = \min \left\{ 3\left(x - \frac{2}{3}\right)^2 + \frac{2}{3} \right\} \tag{11}$$

となるため，$(x, y) = \left(\frac{2}{3}, \frac{1}{3}\right)$ で最小値 $\frac{2}{3}$ をとる．

3.2　双対上昇法の実装に向けて

Lagrange 関数 $\mathcal{L}$ を定義する．

$$\mathcal{L}(x, y, \lambda) = x^2 + 2y^2 + \lambda(x + y - 1) \tag{12}$$

これを使って，式 (10) の主問題の双対問題を考える．$\mathcal{L}(x, y, \lambda)$ は x, y に関しては 2 次関数であり，x, y は独立である．x^2, y^2 の係数は正であるため，最小になる (x, y) は $(x, y) = (-\lambda/2, -\lambda/4)$ と簡単に求められる．そのため，双対問題は式 (12) の x, y にこの値を代入して得られる

$$\phi(\lambda) = \left(-\frac{\lambda}{2}\right)^2 + 2\left(-\frac{\lambda}{4}\right)^2 + \lambda\left(-\frac{\lambda}{2} - \frac{\lambda}{4} - 1\right) = -\frac{3}{8}\lambda^2 - \lambda \tag{13}$$

[1] 式 (7) の otherwise のときは各 λ_i を $+\infty$ または $-\infty$ とすれば，$\max_{\boldsymbol{\lambda} \in \mathbb{R}^p} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ は必ず $+\infty$ となる．

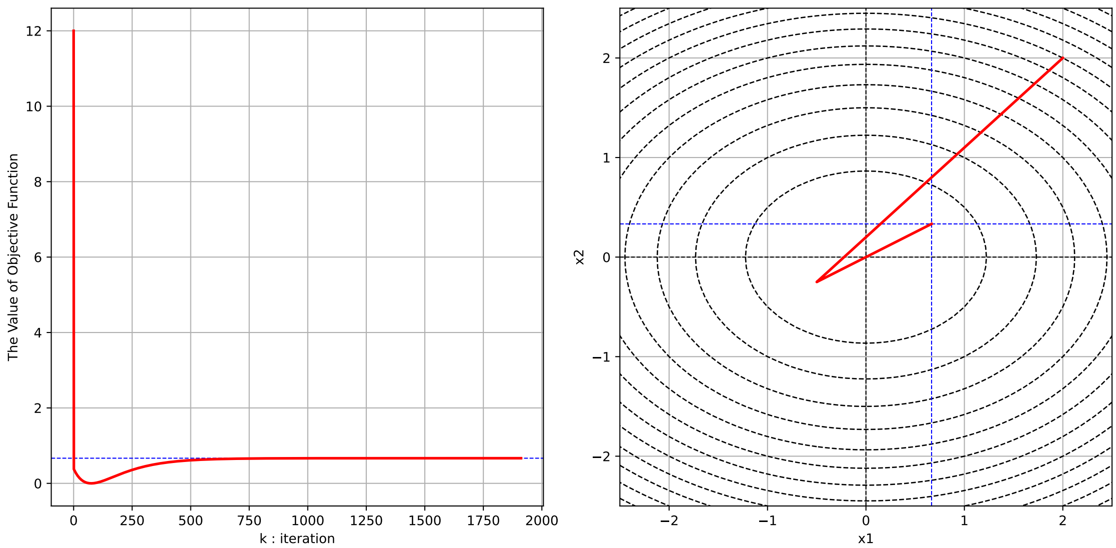

図 1: 実行結果：左図は目的関数 $f(\boldsymbol{x}) = x^2 + 2y^2$ が $\boldsymbol{x}_k = (x_k, y_k)$ の反復更新とともにどのように減少するかを表す．一方，右図は反復更新とともに $\boldsymbol{x}_k$ が 2 次元平面上をどのように移動するかを表す．

の λ に関する最大化問題である．

この $\phi(\lambda)$ であれば最大化は容易である．しかし，実際は主問題よりは簡単になったものも，解析的に最大となる点を求めることは楽ではないことが多い．そこで，適当な最適化手法に基づいて，$\phi(\lambda)$ の最大化を実行する．λ の変化に対応して，$\mathcal{L}(x, y, \lambda)$ が x, y に関して最小となる時の (x, y) も変化する．こうして x, y の最適化も実行される．ここまでの議論から，双対上昇法は次のような流れを実装すればよい．

Step0： 適当に初期値 (x_0, y_0, λ_0) を定める．

Step1： $k = 1, 2, \cdots$ に対して，$(x_{k+1},\ y_{k+1})$ の組を $(x_{k+1},\ y_{k+1}) = \operatorname{argmin}_{x,y\in\mathbb{R}} L(x, y, \lambda_k)$ により決定する．今回の場合は $(x_{k+1}, y_{k+1}) = (-\lambda_k/2, -\lambda_k/4)$ である．

Step2： $\lambda_{k+1} = \lambda_k + \varepsilon_k \phi'(\lambda_k)$ により双対変数 λ の勾配上昇を実行する．$\varepsilon_k > 0$ はステップ幅である．各 k ごとにステップ幅を変えてもよいが，ここでは $\varepsilon_k = 0.01$ で固定することにする．

Step3： Step1 と Step2 を収束するまで繰り返す．今回は収束判定条件を $|\lambda_{k+1} - \lambda_k| < 10^{-8}$ となった時に反復を終了させることにする．

3.3 実装結果

ここまでの情報をもとに Python で，双対上昇法を実装した．その際のソースコードが `dual.py` である．今回は初期点を $\boldsymbol{x}_0 = (2, 2)^\top$ と $\lambda_0 = 1$ に設定した．その際の実行結果が図 1 と図 2 である．収束点は $\boldsymbol{x}^* = (0.6666, 0.3333)^\top$ で，このとき，$f(\boldsymbol{x}^*) = 0.6666$ となった．これは 3.1 節で確認した最小点と最小値と一致する．また，λ の収束値は $\lambda^* = -1.3333$ であった．$\phi(\lambda)$ を最大にする λ は $-4/3$ のため，この値も解析的に得られる解と一致する．

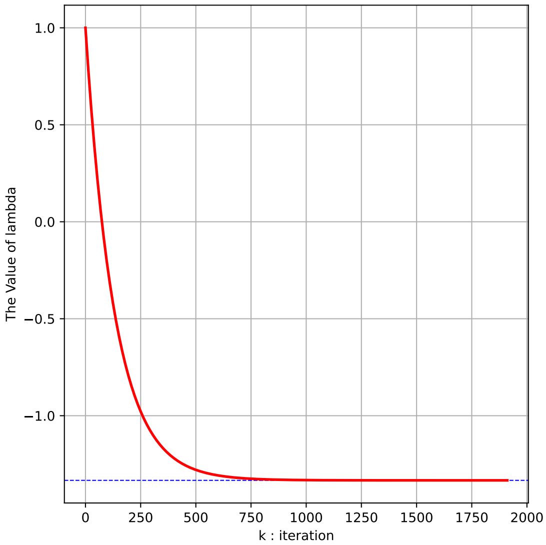

図 2: 実行結果：λ の反復更新に伴う変化

参考文献

[1] 金森敬文，鈴木大慈，竹内一郎，佐藤一誠．機械学習のための連続最適化（第 2 刷）．講談社, 2017

[2] 寒野善博，土谷隆．東京大学工学教程 基礎系 数学 最適化と変分法（第 5 刷）．丸善出版，2018