7.　勾配法のプログラミング (2)

田中雅博

最適化プログラミング、教科書 pp.79–85

多変数関数の極値を数値的に求める代表的な手法その 1 。今回は、多変数の場合。

1　多変数の関数

2 変数関数 $f = f(x, y)$ の最大値を求める問題を考える。1 変数の場合同様、

$$\frac{\partial f}{\partial x} = 0, \quad \frac{\partial f}{\partial y} = 0$$

となるような (x, y) を求める問題であるが、解析的に解けないものとしよう。そのため、勾配法で数値的に求める。

教科書 p.82 の図 3.2 を見よ。k 回目の点 (x_k, y_k) から次の (x_{i+1}, y_{k+1}) に進むところを考えよう。(x_k, y_k) で最も勾配の大きな方向は、この点での勾配ベクトル

$$\nabla f_k = \left(\begin{array}{c} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{array} \right) \Bigg|_{x=x_k, y=y_k}$$

である。

そこで、この方向で関数が極大になる点を探す。これを、**直線探索**という。探索の幅の作り方は、1 次元の場合と同じでよい。

今後の便宜のために、$\boldsymbol{x} = \left(\begin{array}{c} x \\ y \end{array} \right)$ と置こう。

直線探索は、

$$g(t) = f(\boldsymbol{x}_k + t\nabla f_k) \quad (\nabla f_k \text{はベクトルである})$$

とおく。2 次元ベクトルの場合、

$$g(t) = f\left(x_k + t\frac{\partial f}{\partial x_k},\ y_k + t\frac{\partial f}{\partial y_k} \right) \tag{1}$$

となり、最大化は

$$\max_t g(t)$$

と書ける（t をいろいろ変えて、$g(t)$ が最も大きくなるようにする）。

hill_climbing(x_0, y_0)（山登り法）

1. $\boldsymbol{x}$ の初期値を与え、$h \leftarrow h_0$ とする。$k = 0$。
2. $g(t) = f(\boldsymbol{x}_k + t\nabla f_k)$ に対して**直線探索**を行い、$g(t)$ を最大化する t を求める。

3. 得られた t を用いて、今回の結果による、次のスタート地点 $\boldsymbol{x}_{k+1}$ を

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + t\nabla f_k$$

とし、k = k+1 として、Step 2 からここまで、$\|\Delta \boldsymbol{x}_k\| < \delta$ となるまで繰り返す。

□

以下、直線探索のアルゴリズムを示す。探索する軸は、勾配の向きの方向に固定することで、直線探索となる。g は t の関数であることに注意する。

t = linsearch(x_k, y_k)（直線探索）のアルゴリズム

1. t の初期値を与え、$h \leftarrow h_0$ とする。
2. 次のように置く。
$$h \leftarrow \mathrm{sgn}(g'(t))|h|, \quad T \leftarrow t, \quad T' \leftarrow t + h$$
3. もし $g(t) < g(T')$ であれば、次の計算を行う。
   (a) $g(t) \geq g(T')$ となるまで次の計算を繰り返す。
   $$h \leftarrow 2h, \quad T \leftarrow T', \quad T' \leftarrow T + h$$
   (b) $t \leftarrow T, h \leftarrow h/2$ と置く。
4. そうでなければ、次の計算を行う。
   (a) $g(t) \leq g(T')$ となるまで、次の計算を繰り返す。
   $$h \leftarrow \frac{h}{2}, \quad T' \leftarrow T' - h$$
   (b) $t \leftarrow T', h \leftarrow 2h$ と置く。
5. ステップ 2 に戻り、これを $|g'(t)| \leq \epsilon$ となるまで繰り返す。
6. 得られた t を返す。

実行に必要な関数プログラム

- hill_climb（山登り関数の大枠）
- linsearch（線形探索. 山登りの中で使う）
- g（目的関数）
- gp（目的関数の微分）

例題

$$f(x, y) = -(x - y)^2 - (x - 1)^2$$

の最大値を求める。

$$g(t) = -\left(\left(x_k + t\frac{\partial f}{\partial x_k}\right) - \left(y_k + t\frac{\partial f}{\partial y_k}\right)\right)^2 - \left(\left(x_k + t\frac{\partial f}{\partial x_k}\right) - 1\right)^2$$

$$\frac{dg}{dt} = -2\left((x-y) + (x-1)\right)\frac{\partial f}{\partial x_k} + 2(x-y)\frac{\partial f}{\partial y_k}$$

これの MATLAB プログラムを示そう。関数 $g(t)$ を `g(t,x_k,y_k,fx_k,fy_k)`、その導関数 $g'(t)$ を `gp(t,x_k,y_k,fx_k,fy_k)` とする。山登り法を `hill_climbing(x_0,y_0)`、線形探索を `linsearch(x_k,y_k)` とする。

```
%勾配法メインプログラム
function [xrec, yrec] = hill_climbing(x_0, y_0)
delta2 = 1.0e-12;
delx = 1; %初回の while 分を実行させるためのもの
dely = 1; %初回の while 分を実行させるためのもの
count = 1;
x_k = x_0;
y_k = y_0;
dx_k = -2 * (x_k - y_k) - 2*(x_k - 1); % ∂ f/∂ x
dy_k = 2 * (x_k - y_k); % ∂ f/∂ y
xrec(1) = x_k;
yrec(1) = y_k;
%iteration = 0;
while (delx^2 + dely^2) > delta2 && count < 100
    %iteration = iteration + 1;
    count = count + 1;
    t = linsearch(x_k, y_k);
    delx = t * dx_k;
    dely = t * dy_k;
    x_k = x_k + delx;
    y_k = y_k + dely;
    dx_k = -2 * (x_k - y_k) - 2 * (x_k - 1);
    dy_k = 2 * (x_k - y_k);
    disp([x_k y_k]);
    xrec(count) = x_k;
    yrec(count) = y_k;
end
'最適解'
disp([x_k, y_k])
plot(xrec,yrec,'o-')
hold on
x = -10 : 0.2 : 10;
y = -10 : 0.2 : 10;
[xx, yy] = meshgrid(x, y);
z = kansu2(xx, yy);
contour(xx, yy, z)
hold off
end
```

```
function t = linsearch(x_k, y_k)
t=0;
dx_k = -2 * (x_k - y_k) - 2 * (x_k - 1);
dy_k = 2 * (x_k - y_k);
%pause
if gp(0, x_k, y_k, dx_k, dy_k) > 0
    h0 = 0.1;
else
    h0 = -0.1;
end
h = h0;
epsilon = 1.0e-8;
iteration = 0;
while abs(gp(t, x_k, y_k, dx_k, dy_k)) > epsilon && iteration < 100 %探索点のx, y
の座標値はgpの中で計算している。繰り返し回数上限を設定している
    iteration = iteration + 1;
    h = sign(gp(t, x_k, y_k, dx_k, dy_k)) * abs(h);
    T = t;
    Tp = t + h;
    if g(T, x_k, y_k, dx_k, dy_k) < g(Tp, x_k, y_k, dx_k, dy_k)
        %'case1'
        iteration1 = 0;
        while g(T, x_k, y_k, dx_k, dy_k) < g(Tp, x_k, y_k, dx_k, dy_k) && iteration1 < 100
            iteration1 = iteration1 + 1;
            h = h * 2;
            T = Tp;
            Tp = T + h;
        end
        t = T;
        h = h / 2;
    else
        %'case2'
        iteration2  = 0;
        while g(T, x_k, y_k, dx_k, dy_k) > g(Tp, x_k, y_k, dx_k, dy_k)  && iteration2 < 100
            iteration2 = iteration2 + 1;
            h = h / 2;
            Tp = Tp - h;
            %pause
        end
        t = Tp;
        h = 2 * h;
    end
end
```

```
function f = g( t,x_k,y_k,dx_k,dy_k )
x = x_k + t * dx_k;
y = y_k + t * dy_k;
f=-(x-y)^2-(x-1)^2;
end

function dg = gp(t,x_k,y_k,dx_k,dy_k)
x = x_k + t * dx_k;
y = y_k + t * dy_k;
dg = -2*((x-y)+(x-1))*dx_k + 2*(x-y)*dy_k;
end

function z = kansu2(x,y) %対象の関数
z=-(x-y).^2 - (x-1).^2;
end
```

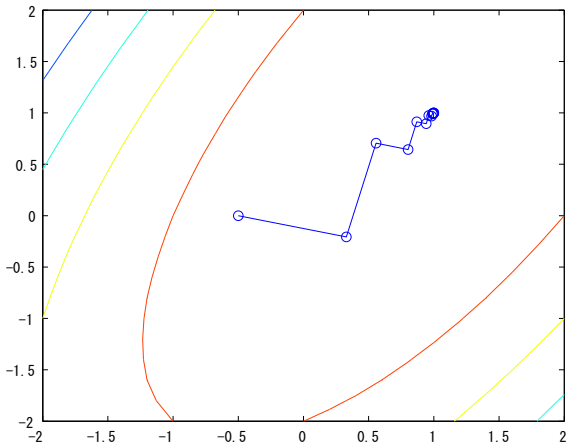
実際に試してみよう。

```
>> hill_climbing(5,5)
>> main
    3.0000    5.0000
    3.0000    3.0000
    2.0000    3.0000
    2.0000    2.0000
    1.5000    2.0000
    1.5000    1.5000
    1.2500    1.5000
    1.2500    1.2500
    1.1250    1.2500
    1.1250    1.1250
    1.0625    1.1250
    1.0625    1.0625
    1.0312    1.0625
    1.0313    1.0312
    1.0156    1.0313
    1.0156    1.0156
    1.0078    1.0156
    1.0078    1.0078
    1.0039    1.0078
    1.0039    1.0039
    1.0020    1.0039
    1.0020    1.0020
    1.0010    1.0020
    1.0010    1.0010
    1.0005    1.0010
    1.0005    1.0005
    1.0002    1.0005
    1.0002    1.0002
    1.0001    1.0002
    1.0001    1.0001
    1.0001    1.0001
    1.0001    1.0001
    1.0001    1.0001


ans =
```

最適解

```
1.0001    1.0001
```



課題 7

1. 上記で与えられた関数の最大化について、いくつかの点からスタートして、最適解に収束するかどうか、試してみよ。
2. 次に、$f(x, y) = -(\cos x - x)^2 - (x - y)^4$ の最大化を行う。
   (a) 必要に応じて、hill_climb 関数、linsearch 関数、g 関数、gp 関数を変更せよ。
   (b) 実行してみて、いくつかの初期値 (x0, y0) とそれぞれの場合得られた最適解を数値で示せ (x, y, f の値)。

[提出物] Word に、以下の内容を示せ。

- 学籍番号、氏名、課題番号 7
- 問 1 について
  - 収束状況（発散すれば、その旨）を示し、それについて考察せよ。
- 問 2 について
  - $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$ を式で書け（手書きでもよい）。
  - MATLAB プログラムリスト
  - 2. のプログラムの中で、1. のプログラムとは違う部分をわかりやすく示せ（プログラムの中の該当する部分をマーカーで塗るか、四角で囲むなど）
  - 収束状況についての考察