**最优化问题**

**一、牛顿法**

**1. 算法原理**

假设任务是优化一个目标函数f，求函数f的极大极小问题，可以转化为求解函数f的导数f'=0的问题，这样求可以把优化问题看成方程求解问题（f'=0）。剩下的问题就和第一部分提到的牛顿法求解很相似了。

这次为了求解f'=0的根，把f（x）的泰勒展开，展开到2阶形式：

http://hi.csdn.net/attachment/201106/5/0_1307264093I733.gif

这个式子是成立的，当且仅当 Δ*x*无线趋近于0。此时上式等价与：

http://hi.csdn.net/attachment/201106/5/0_130726445821TF.gif

求解：

http://hi.csdn.net/attachment/201106/5/0_1307264526620L.gif

得出迭代公式：

http://hi.csdn.net/attachment/201106/5/0_1307264554Ly1S.gif

**2. 程序实现**

由于函数可能不存在可逆Hessian矩阵，所以本实验采用拟牛顿法。即Hessian矩阵不可逆的时候用单位矩阵代替。

实验采用多个初始点，误差系数为10^(-5)，实验函数：

X0 = [1.5, 3]; % initial point

E = 0.00001; % error coefficient

N = 1000; % iterative times

syms x1 x2;

X = [x1, x2]; % variables

y = x1 + x2 / ((x1 \* x2) - 1);

G = jacobian(y, X);

H = hessian(y, X);

yX = zeros(N, 1);

yx = zeros(N, 1);

for k = 1 : N

g = double(subs(G, X, X0));

h = double(subs(H, X, X0));

if(det(h)==0)

h = eye(2);

else

h = inv(h);

end

x(k,:) = X0 - (h \* (g'))';

yX= double(subs(y, X, X0));

yx(k) = double(subs(y, X, x(k,:)));

if((yX - yx(k)) <= E)

break;

else

X0 = x(k,:);

end

end

**3. 实验结果**

|  |  |  |
| --- | --- | --- |
| 初始点(x1, x2) | 收敛点及收敛值(x1, x2, y) | 迭代次数K |
| (2,3) | (0.0134,7.1919,-7.934) 发散 | 2 |
| (1.5, 3) | (1, 71754,2) 收敛 | 25 |
| (0.5, 3) | (1, 67438,2) 收敛 | 29 |
| (0,3) | (-0.1667, 3.1667, -2.2394) 发散 | 1 |
| (-0.5,3) | (-1, 73383,-2) 收敛 | 25 |
| (-1.5,3) | (0.0941, 8.0433, -32.9372) 发散 | 2 |

**4. 结论**

此函数没有极值点，因为X2会发散，函数在（1，∞）处收敛为2；在（-1，∞）处收敛为-2。如果初始值偏离1较大，那么就无法迭代到收敛点。

所以牛顿法对初始点的选取非常重要。

**二、梯度下降法**

**1. 算法原理**

梯度下降法，就是利用负梯度方向来决定每次迭代的新的搜索方向，使得每次迭代能使待优化的目标函数逐步减小。梯度下降法是2范数下的最速下降法。 最速下降法的一种简单形式是：x(k+1)=x(k)-a\*g(k),其中a称为学习速率，可以是较小的常数。g（k）是x(k)的梯度。

**2. 程序实现**

本实验中，学习率a采用两种方案，固定步长和最优步长法。实验函数为

X0 = [-3,1]; % initial point

E = 0.000001; % error coefficient

syms x y;

X = [x, y]; % variables

f = cos(x) \* cos(y);

G = jacobian(f, X);

k = 0;

syms m;

while(1)

g = double(subs(G, X, X0));

x = X0 - m .\* g;

h = diff(subs(f, X, x),'m');

a = double(solve(h,'m'));

x = X0 - a .\* g;

yX = double(subs(f, X, X0));

yx = double(subs(f, X, x));

if(abs(yX - yx) <= E)

break;

else

X0 = x;

end

k = k + 1;

end

**3. 实验结果**

对于固定步长实验，改变学习率a的值对于迭代次数有很大影响，有时甚至永远迭代不到满足要求的点。一般实验结果是收敛为（-3.1416,0），值为-1，由于函数以2π为周期，其他收敛点以此类推

对于可变步长的实验，初始点选取不同对实验结果影响很大

|  |  |  |
| --- | --- | --- |
| 初始点（x, y） | 收敛点和收敛值 | 迭代次数 |
| （-3,1） | （-3.1416，0） -1 | 2 |
| （1,1） | （0，0） 1 | 1 |
| （2,1） | （1.5711，1.5710）0 | 6 |
| （3,1） | （3.1416，0） -1 | 2 |

**4. 结论**

在实验过程中，步长（学习率）a的选取对实验结果非常重要。在固定步长的实验中，当a过大，会永远取不到极值点，因为它会在极值点附近来回跳动。在可变步长的实验中，初始点选取的不好会很容易陷入局部最优