1010602 YiWen Chen 2015.5.10

MLevenberg--Marquardt Modification

Introduction

As we known, high-dimensional problem needs to use gradient function to find the minimizer which exists $\nabla f(X^*) = 0$. The formula can be referred as following,

$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - [D^2 f(\boldsymbol{X}_k)]^{-1} \nabla f(\boldsymbol{X}_k)$.

However, if $D^2 f(\boldsymbol{X}_k)$ which is Hessian matrix is equal to 0, this method will be failed. To solve this situation, the Levenberg–Marquardt algorithm is combined the Gauss–Newton algorithm and the method of gradient descent. We plus u_k*I to Hessian matrix as following,

$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - [D^2 f(\boldsymbol{X}_k) + u_k * I]^{-1} \nabla f(\boldsymbol{X}_k)$, so $[D^2 f(\boldsymbol{X}_k) + u_k * I]^{-1}$ may large than 0.

Example by High-dimensional Newton's Method

I choose 40 + (x-4).*x.^3 + 3.*(y-5).^2 as example, its Hessian matrix is = 0 when initial point is [0 5]. As the result, High-dimensional Newton's Method is failed.

```
xk = [0 5]';                                %initial point
precision = 1;                              %set the precision
syms x;                                     %set my example
syms y;
f =( 40 + (x-4).*x.^3 + 3.*(y-5).^2 );
fxy = gradient(f)                           %gradient of my function
hes = hessian(f,[x,y])                      %hessian matrix of my function
f =@(x,y)( 40 + (x-4).*x.^3 + 3.*(y-5).^2 );
    while precision > 0.0001 && times <200  %condition for finishing the loop
        x = xk(1);
        y = xk(2);
        gra(times,1) = xk(1);
        gra(times,2) = xk(2);
        value = subs(f);
        gra(times,3) = value;
        dir = subs(fxy);
        hes2 = subs(hes)
        xk = xk - inv(hes2)*dir             %newton's method
        diff(times) =
abs(f(gra(times,1),gra(times,2))-f(gra(times-1,1),gra(times-1,2)));
                                            %calculate the difference
        precision = diff(times);
        m(times-1) = times-1;
```

```
        times = times + 1;
    end
```

Figure 1 represets Newton's failed because Hessian matrix is = 0.

```
hes =

[ 6*x*(x - 4) + 6*x^2, 0]
[                   0, 6]


hes2 =

     0     0
     0     6

Warning: Matrix is singular to working precision.
> In highNew at 39

xk =

   NaN
   NaN


hes2 =

   NaN     0
     0     6

Warning: Matrix is singular to working precision.
> In highNew at 39
```

Figure 1output of example by High-dimensional Newton's Method

● Example by Levenberg-Marquardt Modification

```
x = -5:0.01:5;                                    %plot the example
y = 0:0.1:10;

[xx,yy] = meshgrid(x,y);
ff = 40 + (xx-4).*xx.^3 + 3.*(yy-5).^2;
subplot(1,2,1);
hold on;
mesh(xx,yy,ff)
subplot(1,2,2);
hold on;
contour(xx,yy,ff)

xk = [0 5]';                                      %initial point
times = 2;                                        %operation times
precision = 1;
```

```
c = 1;
I = eye(2)

syms x;                                          %set my function
syms y;
f =( 40 + (x-4).*x.^3 + 3.*(y-5).^2 );
fxy = gradient(f)
hes = hessian(f,[x,y])
f =@(x,y)( 40 + (x-4).*x.^3 + 3.*(y-5).^2 );
    while precision > 0.0001 && times <200       %condition for finishing the loop
        x = xk(1);
        y = xk(2);
        gra(times,1) = xk(1);
        gra(times,2) = xk(2);
        value = subs(f);
        gra(times,3) = value;
        dir = subs(fxy);
        hes2 = subs(hes)                         %hessian matrix of my example
        xk = xk - inv(hes2 + c*I)*dir            %L-M modification
        subplot(1,2,1);
        hold on;
        plot3(xk(1),xk(2),value)                 %plot in left picture
        subplot(1,2,2);
        hold on;
        plot3(xk(1),xk(2),value)                 %plot in right picture
        diff(times) =
abs(f(gra(times,1),gra(times,2))-f(gra(times-1,1),gra(times-1,2)));
        precision = diff(times);                 %calculate the difference
        m(times-1) = times-1;
        times = times + 1;
    end
times
```

By L-M modification, my initial point is [0 5], step size is 1. As shown in Figure 2, x* = 0, y* =5 and f(x*, y*) = 40 through iterated 1 times.

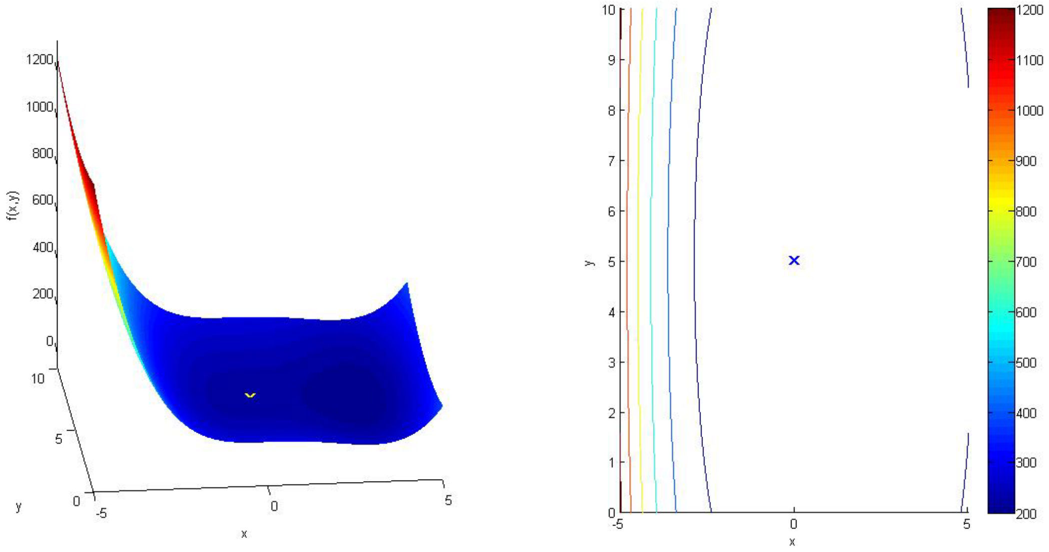

Figure 2output of example by Levenberg-Marquardt Modification

By L-M modification, my initial point is [2 5], step size is 1. As shown in Figure 2, x* = 3, y* =5 and f(x*, y*) = 13 through iterated 10 times.

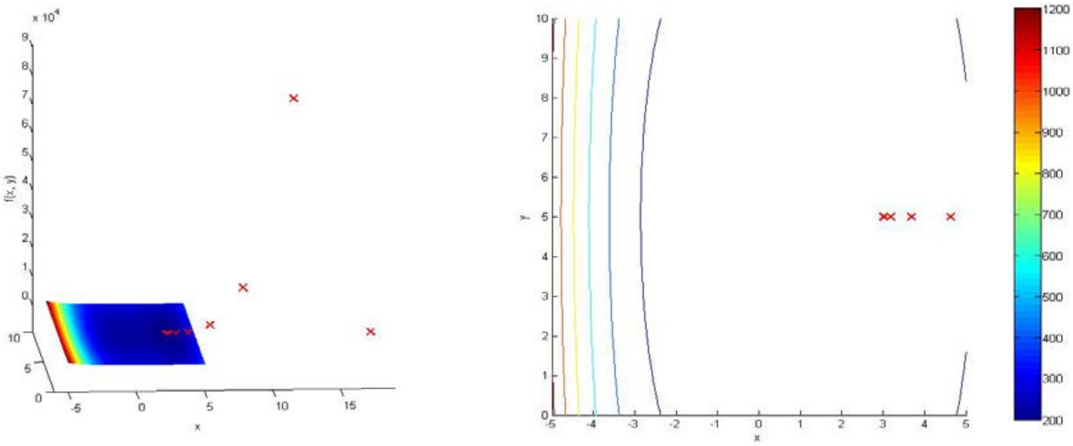

Figure 3output of example by Levenberg-Marquardt Modification

As the result, my example has two critical point in [0 5] and [3 5], shown in Figure 4.

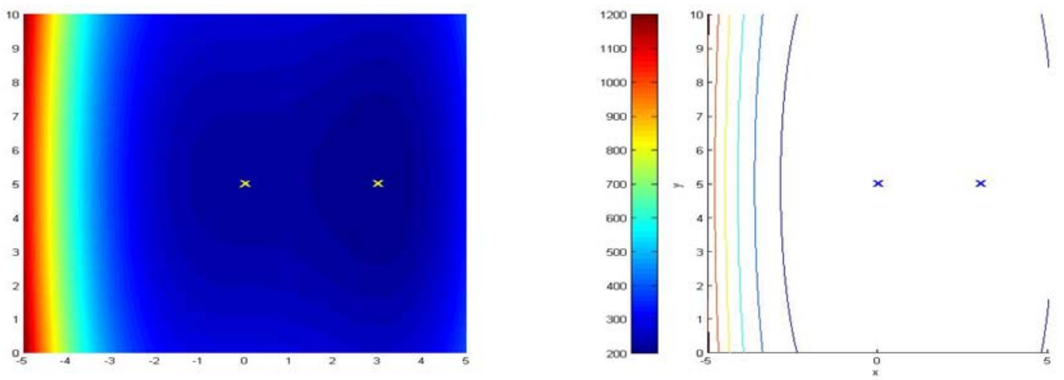

Figure 4 critical point