优化实用算法: 第五次作业

2022 年 5 月 28 日

指导老师：徐翔

徐圣泽 3190102721

Problem 1

1. Let $r_1(x) = x_2 - x_1^2$, $r_2(x) = 1 - x_2$, $r(x) = (r_1(x), r_2(x))^T$, $f(x) = \frac{1}{2}[r_1(x)^2 + r_2(x)^2] = \frac{1}{2}r(x)^T r(x)$. It is known that the solution to min $f(x)$ is $x^\star = (1,1)^T$.
(a) For any initial guess x0, give the Newton algorithm of

$$\begin{cases} x_{k+1} = x_k + \alpha_k p_k, \\ p_k = -[\nabla^2 f(x)]\nabla f(x), \\ \alpha_k : f(x_k + \alpha_k p_k) = \min f(x + \alpha p_k). \end{cases}$$

(b) Try to prove when $x \to x^\star$,

$$\nabla^2 f(x) \to \nabla r(x) \nabla r(x)^T.$$

解:
(a) 记 $J(x) = \nabla r(x)^T = (\nabla r_1(x), \nabla r_2(x))^T$,则函数 $f(x)$ 的梯度为 $\nabla f(x) = r_1(x)\nabla r_1(x) + r_2(x)\nabla r_2(x) = J(x)^T r(x)$,对应的 $Hesse$ 矩阵为 $\nabla^2 f(x) = \nabla r_1(x)\nabla r_1(x)^T + \nabla r_2(x)\nabla r_2(x)^T + r_1(x)\nabla^2 r_1(x) + r_2(x)\nabla^2 r_2(x) = J(x)^T J(x) + S(x)$。
其中

$$\nabla^2 r_1(x) = \begin{pmatrix} -2 & 0 \\ 0 & 0 \end{pmatrix}, \nabla^2 r_2(x) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, S(x) = \begin{pmatrix} -2r_1(x) & 0 \\ 0 & 0 \end{pmatrix}$$

该问题需要给出牛顿法,与第一次作业的算法要求基本一致,下面进行编程求解。
首先写出函数及其梯度和 $Hesse$ 矩阵。

```
function y=f(x)
r=[x(2)-x(1)^2;1-x(2)];
y=r'*r/2;
end
```

```
function y=Gradf(x)
r=[x(2)-x(1)^2;1-x(2)];
J=[-2*x(1),1;0,-1];
y=J'*r;
end
```

```
function y=Hessef(x)
J=[-2*x(1),1;0,-1];
y=J'*J+[-2*(x(2)-x(1)^2),0;0,0];
end
```

下面我们首先根据书本 57 至 58 页的算法编写了利用进退法进行一维搜索确定区间的函数:

```
function [a,b]=Range(f,a0,h0,x,p)
t=2;k=0;ak=a0;yk=f(x+ak*p);h=h0;ak1=0;yk1=0;alpha=0;
while(0<=k)
        ak1=ak+h;
        if(ak1<0)
```

```
                ak1=0;
                break;
        end
        yk1=f(x+ak1*p);
        if(yk1<=yk)
                h=t*h; alpha=ak; ak=ak1; yk=yk1; k=k+1;
        else
                if k==0
                        h=-h; ak=ak1; yk=f(x+ak*p);
                else
                        break;
                end
        end
end
if(alpha<ak1)
        a=alpha;
        b=ak1;
else
        a=ak1;
        b=alpha;
end
end
```

下面利用 0.618 法进行精确一维搜索：

```
function y = Minimum(f,a0,b0,x,p,epsilon)
a=a0;b=b0;
lambda=a+0.382*(b-a);mu=a+0.618*(b-a);
y1=f(x+lambda*p);y2=f(x+mu*p);
k=1;y=0;
while(k<=10000)
        if(y1>y2)
                if(b-lambda<=epsilon)
                        y=mu;
                        break;
                end
                a=lambda; lambda=mu;
                y1=f(x+lambda*p);
                mu=a+0.618*(b-a);
                y2=f(x+mu*p);
        else
                if(mu-a<=epsilon)
                        y=lambda;
                        break;
                end
                b=mu;mu=lambda;
                y2=f(x+mu*p);
```

```
                lambda=a+0.382*(b-a);
                y1=f(x+lambda*p);
        end
        k=k+1;
end
end
```

下面是带线搜索牛顿法的函数：

```
function [x,k]=NewtonsLineSearch(f,Gradf,Hessef,x0,epsilon1,epsilon2)
k=0;x=x0;
while(k>=0)
        if(sqrt((Gradf(x))'*Gradf(x))<=epsilon1)
                break;
        end
        p=-inv(Hessef(x))*Gradf(x);
        a0=2;h0=0.5;
        [a,b] = Range(f,a0,h0,x,p);
        x=x+Minimum(f,a,b,x,p,epsilon2)*p;
        k=k+1;
end
end
```

下面是主程序的代码：

```
x0=unifrnd(-1000,1000,2,1);
epsilon=10^(-12);
[x,k] = NewtonsLineSearch(@f,@Gradf,@Hessef,x0,epsilon,epsilon);
```

下面是三次随机选取初值后运行的结果：

```
随机选定的初值：
600.5609
-716.2273

带线搜索牛顿法得到的函数极小值点：
-1.0000
1.0000

带线搜索牛顿法得到的函数极小值：
2.4850e-26
```

```
随机选定的初值：
389.6572
-365.8010

带线搜索牛顿法得到的函数极小值点：
```

```
-1
1

带线搜索牛顿法得到的函数极小值：
0

迭代次数为：
12
```

```
随机选定的初值：
-446.1540
-907.6572

带线搜索牛顿法得到的函数极小值点：
1.0000
1.0000

带线搜索牛顿法得到的函数极小值：
8.4926e-30

迭代次数为：
12
```

上面的运行结果说明 $x^\star = (1,1)^T$ 和 $x^\star = (-1,1)^T$ 均为 $f(x)$ 的全局最小值点，此时 $r_1(x^\star) = r_2(x^\star) = f(x^\star) = 0$，故本方法在给定初值不同的情况下给出的极小值点可能不同。
（b）由（a）知，$r_1(x^\star) = r_2(x^\star) = 0$，且 $r_i(x^\star), i = 1,2$ 关于 x 连续，故 $x \to 0$ 时 $r(x) \to 0$，则有

$$\nabla^2 f(x) = J(x)^T J(x) + S(x) = J(x)^T J(x) + \begin{pmatrix} -2r_1(x) & 0 \\ 0 & 0 \end{pmatrix} \to J(x)^T J(x)$$

则有 $\nabla^2 f(x) \to \nabla r(x) \nabla r(x)^T$，即证。

Problem 2

2. Let x_1, x_2 are solutions to $(A^TA + \mu_i I)x = -A^T r, i = 1,2$ with respect to μ_1 and μ_2, where $\mu_1 > \mu_2 > 0, A \in R^{m\times n}, r \in R^m$. Try to prove $\|Ax_2 + r\|_2^2 < \|Ax_1 + r\|_2^2$.
解：对 A^TA 应用 SVD 分解，则有正交矩阵 Q 和对角化矩阵 Λ，使得 $A^TA = Q\Lambda Q^T$，其中 $Q = (q_1, q_2, \cdots, q_n) \in R^{n\times n}$，$\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_n) \in R^{n\times n}$。
根据第四次作业第三道题目的结论，我们有

$$x(\mu_i) = -\sum_{k=1}^{n} \frac{q_k^T A^T r}{\mu_i + \lambda_k} q_k$$

因此有 $\|Ax_i + r\|_2^2 = (Ax_i + r)^T(Ax_i + r) = x_i^T A^T A x_i + x_i^T A^T r + r^T A x_i + r^T r$。

$$\begin{aligned} x_i^T A^T A x_i &= (\sum_{k=1}^{n} \frac{q_k^T A^T r}{\mu_i + \lambda_k} q_k^T) A^T A (\sum_{k=1}^{n} \frac{q_k^T A^T r}{\mu_i + \lambda_k} q_k) \\ &= \sum_{l=1}^{n} \sum_{k=1}^{n} (\frac{q_k^T A^T r}{\mu_i + \lambda_k}) q_k^T A^T A q_l (\frac{q_l^T A^T r}{\mu_i + \lambda_l}) \end{aligned}$$

易知其中

$$q_k^T A^T A q_l = \begin{cases} \lambda_j, & k = l = \lambda_j \\ 0, & k \neq l \end{cases}$$

因此 $x_i^T A^T A x_i = \sum_{k=1}^n (\frac{q^T A^T r}{\mu_i + \lambda_k})^2 \lambda_k$，$x_i^T A^T r = (r^T A x_i)^T = r_T A x_i = -\sum_{k=1}^n \frac{(r^T A q_k)(q_k^T A^T r)}{\mu_i + \lambda_k}$，
故 $\|Ax_i + r\|_2^2 = \sum_{k=1}^n [\frac{\lambda_k (q_k^T A^T r)^2}{(\mu_i + \lambda_k)^2} - \frac{2(q_k^T A^T r)^2}{\mu_i + \lambda_k}] + r^T r$。
对 μ_i 求导，有

$$\begin{aligned} \frac{d}{d\mu_i}[\frac{\lambda_k (q_k^T A^T r)^2}{(\mu_i + \lambda_k)^2} - \frac{2(q_k^T A^T r)^2}{\mu_i + \lambda_k}] &= (q_k^T A^T r)^2 [\frac{-2\lambda_k}{(\mu_i + \lambda_k)^3} + \frac{2}{(\mu_i + \lambda_k)^2}] \\ &= (q_k^T A^T r)^2 \frac{2\mu_i}{(\mu_i + \lambda_k)^3} > 0 \end{aligned}$$

因此 $\|Ax(\mu_i) + r\|_2^2$ 随 μ_i 的增大而增大，因此有 $\|Ax_2 + r\|_2^2 < \|Ax_1 + r\|_2^2$。