第三次大作业

计试 001 苏悦馨 2204120515

1 问题简述

等式约束熵极大化问题

$$\begin{aligned} \min \quad & f(x) = \sum_{i=1}^{n} x_i log x_i \\ s.t. \quad & Ax = b \end{aligned}$$

其中 $dom\ f = R^n_{++}, A \in R^{p\times n}, p < n, rank(A) = p$，问题实例为 $n = 100, p = 30$

随机生成一个满秩矩阵 A，随机选择一个正向量作为 x_0，然后令 $b = Ax_0$ 采用以下方法计算该问题的解

a) 标准 $Newton$ 方法，初始点为 x_0

b) 不可行初始点 $Newton$ 方法，初始点为 x_0, 以及随机生成的向量 v_0

c) 对偶 $Newton$ 方法

2 问题分析

首先随机生成满秩矩阵 A 和正向量 x_0

```
%随机生成满秩矩阵A
A=rand(p,n);
while rank(A) ≠ p
    A=0.1+rand(p,n);
end
%随机生成正向量 x0
x0=rand(n,1);
b=A*x0;  %x0是可行的
```

使用 $matlab$ 符号函数计算 $\nabla^2 f, \nabla f, r(x, v)$

```
%生成原函数以及其他计算中用到的函数
syms X [n,1] matrix
syms V [p,1] matrix
f=X.'* log(X);%原函数f
H_f=diff(f,X,X.');%函数f的hessian矩阵
G_f=diff(f,X)';%函数f的梯度
r=[G_f+A'*V;A*X-sym(b)];%原对偶残差
```

2.1 标准 *Newton* 方法

由可行初始点的标准 *Newton* 方法，下降方向 d_x^k 满足

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} d_x^k \\ w \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ 0 \end{bmatrix}$$

牛顿减少量 $\lambda^2(x^k)$ 满足

$$\lambda^2(x^k) = {d_{nt}^k}^T \nabla^2 f(x) d_{nt}^k$$

实现代码如下

```
value_G_f=double(sym(subs(G_f,x)));
value_H_f=double(sym(subs(H_f,x)));
matrix_A=[value_H_f,A';A,zeros(p,p)];
matrix_b=[-value_G_f;zeros(p,1)];
dr=matrix_A\matrix_b;
d=dr(1:100);%新的牛顿方向
w=dr(101:130);%新的对偶变量
```

回溯直线搜索部分如下

- 外循环：判断 $\frac{1}{2}\lambda^2(x^k) < 10^{-10}$
- 内循环：通过回溯直线搜索求解 t^k，使得 $f(x^k + t^k d^k) < f(x^k) - \alpha t^k \lambda^2(x^k)$

```
while double(sym(subs(f,x_))) > (double(sym(subs(f,x)))-alpha*t*lamda_2)
    t=beta*t;
    x_=x+t*d;
end
```

- 在每次内循环结束后，更新 $x^{k+1}, \nabla f(x^{k+1}), \nabla^2 f(x^{k+1}), \lambda^2(x^{k+1})$

```
%更新参量
x=x+t*d;%新的x

value_G_f=double(sym(subs(G_f,x)));
value_H_f=double(sym(subs(H_f,x)));
matrix_A=[value_H_f,A';A,zeros(p,p)];
matrix_b=[-value_G_f;zeros(p,1)];
dr=matrix_A\matrix_b;
d=dr(1:100);%新的牛顿方向
w=dr(101:130);%新的对偶变量
lamda_2=d'*value_H_f*d;%新的牛顿减少量
```

2.2 不可行初始点 *Newton* 方法

由不可行初始点的标准 *Newton* 方法，下降方向 d_x^k 满足

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} d_x^k \\ d_v^k \end{bmatrix} = -\begin{bmatrix} \nabla f(x) + A^T v \\ Ax^k - b \end{bmatrix}$$

实现代码如下

```
value_G_f=double(sym(subs(G_f,x)));
value_H_f=double(sym(subs(H_f,x)));
matrix_A=[value_H_f,A';A,zeros(p,p)];
matrix_b_2=-[value_G_f+A'*v;A*x-b];
dr_2=matrix_A\matrix_b_2;
dx=dr_2(1:100);%牛顿方向
dv=dr_2(101:130);%对偶变量更改量
```

对原对偶残差 $\|r\|_2$ 回溯部分如下

- 外循环：判断 $\frac{1}{2}\lambda^2(x^k) < 10^{-10}$
- 内循环：通过回溯直线搜索求解 t^k，使得 $f(x^k + t^k d^k) < f(x^k) - \alpha t^k \lambda^2(x^k)$

```
while norm(double(sym(subs(r,{X,V},{x_,v_})))) > (1-alpha*t)*norm_r
    t=beta*t;
    x_=x+t*dx;
    v_=v+t*dv;
end
```

- 在每次内循环结束后，更新 $x^{k+1}, \nabla f(x^{k+1}), \nabla^2 f(x^{k+1}), \|r(x^{k+1}, v^{k+1})\|_2$

```
%更新参量
x=x+t*dx;%新的x
v=v+t*dv;% 新的v

value_G_f=double(sym(subs(G_f,x)));
value_H_f=double(sym(subs(H_f,x)));
matrix_A=[value_H_f,A';A,zeros(p,p)];
matrix_b_2=-[value_G_f+A'*v;A*x-b];
value_r=double(sym(subs(r,{X,V},{x,v})));
norm_r=norm(value_r);%初始点处原对偶残差的范数
dr_2=matrix_A\matrix_b_2;
dx=dr_2(1:100);%牛顿方向
dv=dr_2(101:130);%对偶变量更改量
```

2.3 对偶 *Newton* 方法

首先求出该问题的对偶问题，记 $A^{p\times n} = [a_1, a_2, \cdots, a_n]$
该问题的 *lagrange* 函数为

$$L(x, v) = \sum_{i=1}^{n} x_i log x_i + v^T(Ax - b)$$

由于 *lagrange* 函数为凹函数，则 $\frac{\partial L(x,v)}{\partial x_i} = 0,\ i = 1, 2, \cdots, n$ 时，$g(v) = \inf_x L(x, v)$
由

$$\frac{\partial L(x, v)}{\partial x_i} = \log x_i + 1 + v^T a_i$$

解得

$$x_i = e^{-(1+v^T a_i)},\ i = 1, 2, \cdots, n$$

则代入 x_i，得到对偶函数为

$$g(v) = -e^{-1} \sum_{i=1}^{n} e^{-v^T a_i} - b^T v$$

则得到原问题的对偶问题为

$$\min_{v \in R^p}\ e^{-1} \sum_{i=1}^{n} e^{-v^T a_i} + b^T v$$

对于该无约束优化问题，使用 $Newton$ 下降方法如下（记 $f(v) = -g(v)$）
由牛顿方法，下降方向 d_{nt}^k 满足：

$$d_{nt}^k = -(\nabla^2 f(v^k))^{-1} \nabla f(v^k)$$

牛顿减少量 $\lambda^2(x^k)$ 满足

$$\lambda^2(x^k) = {d_{nt}^k}^T \nabla^2 f(x) d_{nt}^k$$

其中

$$\nabla f(v) = b - A e^{-1-A^T v}$$
$$\nabla^2 f(v) = A\ diag(e^{-A^T v - 1})\ A$$

代码思路:

a) 使用 matlab 符号函数（syms）定义 $-g(v)$，加负号是为了变成凸优化问题。

```
g=sym(ones(1,n))*exp(-sym(ones(n,1))-sym(A')*V)+sym(b')*V;
```

b) 外循环：判断 $\frac{1}{2}\lambda^2(v^k) < 10^{-10}$

c) 内循环：通过回溯直线搜索求解 t^k，使得 $f(v^k + t^k d^k) < f(v^k) - \alpha t^k \lambda^2(v^k)$

```
while double(sym(subs(g,v_))) > (double(sym(subs(g,v)))-alpha*t*lamda_v)
      t=beta*t;
      v_=v+t*d_v;
end
```

d) 在每次内循环结束后，记录 $f(v^k) - p^*$，更新 $v^{k+1}, \lambda^2(v^k)$

```
    v=v+t*d_v;
    g_v=b-A*exp(-1-A'*v);
    hessian_v=A*diag(exp(-1-A'*v))*A';
    d_v = -hessian_v\g_v;%下降方向
    lamda_v= d_v'*hessian_v* d_v; %牛顿减少量的平方
    k=k+1;
```

3 结果分析

3.1 证实求得相同的最优点

运行程序后分别保存求得的最优变量 x, v 的值（迭代中不涉及求某一变量求值的，代入 x, v 的相关表达式计算），后通过计算 $\|x_i - x_j\|_2$, $i, j = a, b, c$, $\|v_i - v_j\|_2$, $i, j = a, b, c$ 来验证是否求得了相同的最优点。

代码如下

```
    norm(x_a-x_b)
    norm(x_b-x_c)
    norm(x_c-x_a)

    norm(v_a-v_b)
    norm(v_b-v_c)
    norm(v_c-v_a)
```

最优解之间的 $Euclid$ 距离为（对应上述程序）
$[9.0099e-06, 1.4022e-06, 9.0982e-06, 2.4200e-11, 9.9998e-07, 9.9998e-07]$
由于计算误差的存在，最优解之间极小的的 $Euclid$ 距离是被允许的，从而可证实求得了相同的最优点。

3.2 迭代次数比较

方法	迭代次数
标准 $Newton$ 方法	6
不可行初始点 $Newton$ 方法	8
对偶 $Newton$ 方法	9

可以看出，使用不同的方法求解该问题，在从同一初始点出发，到达指定的精度的情况下，迭代次数并没有太大的差别。