October 10, 2018

Empirical Methods HA 3

Konstantin Guryev

Pennsylvania State University

2018

Problem 1

$$\beta_{mle} = \begin{pmatrix} 2.5339 \\ -0.0323 \\ 0.1157 \\ -0.3540 \\ 0.0798 \\ -0.4094 \end{pmatrix};$$

Problem 2

$$\beta_{mleqn} = \begin{pmatrix} 2.5339 \\ -0.0323 \\ 0.1157 \\ -0.3540 \\ 0.0798 \\ -0.4094 \end{pmatrix};$$

Problem 3

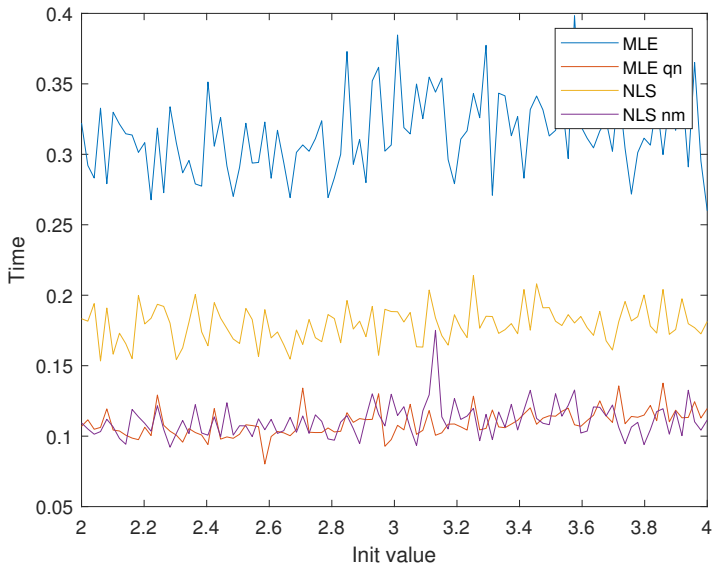
$$\beta_{nls} = \begin{pmatrix} 2.5126 \\ -0.0384 \\ 0.1141 \\ -0.2796 \\ 0.0676 \\ -0.3698 \end{pmatrix};$$

Problem 4

$$\beta_{nlsnm} = \begin{pmatrix} 2.5126 \\ -0.0384 \\ 0.1141 \\ -0.2796 \\ 0.0676 \\ -0.3698 \end{pmatrix};$$

Problem 5

By looking on the graph we may conclude that MLE via the Nelder-Mead is the slowest method, then comes the NLS (lsqonlin) method. And MLE via a quasi-Newton and NLS via Nelder-Mead are the fastest ones.



Matlab Code

```
function [L, J] = likelihood1(y,x,b)
s = size(x);
k1 = y-exp(x*b);
k2 = kron(k1,ones(1,s(2)));
z1 = k2.*x;
J = sum(z1,1)'/s(1);
L1 = -exp(x*b)+y.*x*b;
L = sum(L1)/s(1);
end
```

Matlab Code

```
function [L, J] = likelihood2(y,X,b)
s = size(X);
k1 = y-exp(X*b);
k2 = kron(k1,ones(1,s(2)));
z1 = k2.*X;
L1 = -exp(X*b)+y.*X*b;
L = norm(sum(L1));
J = sum(z1,1)'/(2*L);
end
```

Matlab Code

```
function [a, b, c] = mle_est(init,y,X)
b_mle = init;
l = @(b)-likelihood1(y,X,b);
opt1 = optimset('Display','Final','TolF',1e-8,'TolX',1e-8);
exit = 2;
maxit = 0;
tic
while (exit~=1)&&(maxit<100)
    [b_mle,fval,exit] = fminsearch(l,b_mle,opt1);
    maxit = maxit+1;
end
a = toc;
b = exit;
c = 1;
end
```

Matlab Code

```
function [a, b, c] = mle_qn_est(init,y,X)
b_mle_init = init;
opt2 = optimset('Display','Final');
l_z = @(b)likelihood2(y,X,b);
tic
[b_mle_qn,fval,exit] = fminunc(l_z,b_mle_init,opt2);
a = toc;
b = exit;
c = 2;
end
```

Matlab Code

```
function [a, b, c]=nls_est(init,y,X)
b_nls_init = init;
opt3 = optimoptions('lsqnonlin','Display','Final','TolF',1e-16,'TolX',1e-16);
f=@(b)y-exp(X*b);
tic
[b_nls,res1,res2,exit]=lsqnonlin(f,b_nls_init,[],[],opt3);
a = toc;
b = exit;
c = 3;
end
```

Matlab Code

```
function [a, b, c]=nls_nm_est(init,y,X)
b_nls_nm = init;
nls_f = @(b)sum((y-exp(X*b)).^2);
opt4 = optimset('Display','Final','TolF',1e-16,'TolX',1e-16);
exit2 = 2;
maxit2 = 0;
tic
while (exit2~=1)&&(maxit2<100)
    [b_nls_nm,fval,exit2] = fminsearch(nls_f,b_nls_nm,opt4);
    maxit2 = maxit2+1;
end
a = toc;
b = exit2;
c = 4;
end
```

Matlab Code

```
1
2  M=load('hw3','-mat');
3  X = M.X;
4  y = M.y;
5  clear M;
6
7  %% Problem 1
8  b_mle = [3;0;0;0;0;0];
9  l = @(b)-likelihood1(y,X,b);
10 options = optimset('Display','Final','TolF',1e-8,'TolX',1e-8);
11 exit1 = 2;
12 maxit1 = 0;
13 while (exit1~=1)&&(maxit1<100)
14     [b_mle,fval,exit1]=fminsearch(l,b_mle,options);
15     maxit1 = maxit1+1;
16 end
17 fprintf('Iter: %d, Likelihood f value: %f \n', maxit1,likelihood1(y,X
       ,b_mle));
18 b_mle
19
20 %% Problem 2
21 b_mle_init = [3;0;0;0;0;0];
22 options2 = optimset('Display','final');
23 l_z=@(b)likelihood2(y,X,b);
24 b_mle_qn = fminunc(l_z,b_mle_init,options2);
25 fprintf('Norm of Likelihood f value: %f \n', likelihood2(y,X,
       b_mle_qn));
26 b_mle_qn
27
28 %% Problem 3
29 b_nls_init = [3;0;0;0;0;0];
30 options3 = optimoptions('lsqnonlin','Display','Final','TolF',1e-16,'
       TolX',1e-16);
31 f=@(b)y-exp(X*b);
```

```
b_nls = lsqnonlin(f,b_nls_init,[],[],options3);
fprintf('Obj Function: %f \n',sum(f(b_nls).^2));
b_nls

%% Problem 4
b_nls_nm = [3;0;0;0;0;0];
nls_f=@(b)sum((y-exp(X*b)).^2);
options4 = optimset('Display','Final','TolF',1e-16,'TolX',1e-16);
exit2 = 2;
maxit2=0;
while (exit2~=1)&&(maxit2<100)
    [b_nls_nm,fval,exit2]=fminsearch(nls_f,b_nls_nm,options4);
    maxit2 = maxit2+1;
end
fprintf('Iter: %d, Obj Function: %f \n', maxit2,nls_f(b_nls_nm));
b_nls_nm

%% Problem 5

range = linspace(2,4);
tests = zeros(length(range)*4,4);
for k = 1:length(range)
    b_init = [range(k);0;0;0;0;0];
    [x1,x2,x3] = mle_est(b_init,y,X);
    [x4,x5,x6] = mle_qn_est(b_init,y,X);
    [x7,x8,x9] = nls_est(b_init,y,X);
    [x10,x11,x12] = nls_nm_est(b_init,y,X);
    tests((k-1)*4+1,1) = range(k);
    tests((k-1)*4+1,2) = x1;
    tests((k-1)*4+1,3) = x2;
    tests((k-1)*4+1,4) = x3;
    tests((k-1)*4+2,1) = range(k);
    tests((k-1)*4+2,2) = x4;
    tests((k-1)*4+2,3) = x5;
    tests((k-1)*4+2,4) = x6;
```

```
67      tests((k-1)*4+3,1) = range(k);
68      tests((k-1)*4+3,2) = x7;
69      tests((k-1)*4+3,3) = x8;
70      tests((k-1)*4+3,4) = x9;
71      tests((k-1)*4+4,1) = range(k);
72      tests((k-1)*4+4,2) = x10;
73      tests((k-1)*4+4,3) = x11;
74      tests((k-1)*4+4,4) = x12;
75  end
76  z1 = find(tests(:,4)==1);
77  z2 = find(tests(:,4)==2);
78  z3 = find(tests(:,4)==3);
79  z4 = find(tests(:,4)==4);
80  mle_tests = tests(z1,:);
81  mleqn_tests = tests(z2,:);
82  nls_tests = tests(z3,:);
83  nlsnm_tests = tests(z4,:);
84  plot(range,mle_tests(:,2),range,mleqn_tests(:,2),range,nls_tests
        (:,2),range,nlsnm_tests(:,2));
85  xlabel('Init value');
86  ylabel('Time');
87  legend('MLE','MLE qn','NLS','NLS nm');
```