ECON 512
Homework 3

Pin Sun*

October 10, 2018

Problem 1

```
clear;
close all;
dat=load('hw3.mat');
numy=length(dat.y);
fun=@(beta)-sum(-exp(dat.X(:,1)*beta(1)+dat.X(:,2)*beta(2)+ ...
dat.X(:,3)*beta(3)+dat.X(:,4)*beta(4)+dat.X(:,5)*beta(5) ...
+dat.X(:,6)*beta(6))+dat.y(:,1).*(dat.X(:,1)*beta(1)+dat.X(:,2)*beta(2)+ ...
dat.X(:,3)*beta(3)+dat.X(:,4)*beta(4)+dat.X(:,5)*beta(5) ...
+dat.X(:,6)*beta(6)))/numy;
%options = optimset('PlotFcns',@optimplotfval, 'Display','iter');
beta0 = 1/10*ones(6,1);
tic
beta=fminsearch(fun,beta0)
%beta=fminsearch(fun,beta0,options)
toc
beta =

2.5339
-0.0323
0.1157
-0.3540
0.0798
-0.4094

Elapsed time is 0.642027 seconds.
```

*pxs251@psu.edu

Problem 2

```
-0.4094

Elapsed time is 0.642027 seconds.
>> % Broyden Method
clear;
close all;
dat=load('hw3.mat');
beta = 1/100*ones(6,1);
fVal=loglike(dat.y,dat.X,beta')
iJac=inv(myJac('loglike',dat.y,dat.X,beta'));
maxit = 100;
tol = 1e-6;
tic
for iter = 1:maxit
fnorm = norm(fVal);
if norm(fVal) < tol
break
end
d = - (iJac * fVal);
beta = beta+d;
fOld = fVal;
fVal = loglike(dat.y,dat.X,beta');
u = iJac*(fVal - fOld);
iJac = iJac + ( (d - u) * (d'*iJac) )/ (d'*u);
end
toc
beta

fVal =

-0.2598
-7.8381
0.0563
-1.4499
-0.9065
-1.9989

Elapsed time is 0.044496 seconds.

beta =

2.5339
-0.0323
0.1157
-0.3540
```

```
0.0798
-0.4094
```

Problem 3

```
% Computation Method: trust-region-reflective algorithm
clear;
close all;
dat=load('hw3.mat');
numy=length(dat.y);
fun=@(beta)(-exp(dat.X(:,1)*beta(1)+dat.X(:,2)*beta(2)+ ...
dat.X(:,3)*beta(3)+dat.X(:,4)*beta(4)+dat.X(:,5)*beta(5) ...
+dat.X(:,6)*beta(6))+dat.y(:,1))/numy;
%options = optimset('PlotFcns',@optimplotfval, 'Display','iter');
beta0 =  1/100*ones(6,1);
tic
beta=lsqnonlin(fun,beta0)
toc

Local minimum possible.

lsqnonlin stopped because the final change in the sum of squares relative to
its initial value is less than the default value of the function tolerance.

<stopping criteria details>


beta =

2.5128
-0.0384
0.1141
-0.2796
0.0676
-0.3695

Elapsed time is 1.152794 seconds.
```

Problem 4

```
clear;
close all;
dat=load('hw3.mat');
numy=length(dat.y);
fun=@(beta)sum((-exp(dat.X(:,1)*beta(1)+dat.X(:,2)*beta(2)+ ...
dat.X(:,3)*beta(3)+dat.X(:,4)*beta(4)+dat.X(:,5)*beta(5) ...
+dat.X(:,6)*beta(6))+dat.y(:,1)).^2)/numy;
```

```
%options = optimset('PlotFcns',@optimplotfval, 'Display','iter');
beta0 =1/100*ones(6,1);
tic
beta=fminsearch(fun,beta0)
%beta=fminsearch(fun,beta0,options)
toc

beta =

2.5125
-0.0384
0.1141
-0.2796
0.0677
-0.3698

Elapsed time is 0.376051 seconds.
```

Problem 5

```
clear;
close all;
a=0.01:0.1:.51;
beta1=zeros(length(a),6);
beta2=beta1;
beta3=beta1;
beta4=beta1;
Tm=zeros(4,length(a));
for i=1:length(a)
dat=load('hw3.mat');

% First Approach

numy=length(dat.y);
fun1=@(beta)-sum(-exp(dat.X(:,1)*beta(1)+dat.X(:,2)*beta(2)+ ...
dat.X(:,3)*beta(3)+dat.X(:,4)*beta(4)+dat.X(:,5)*beta(5) ...
+dat.X(:,6)*beta(6))+dat.y(:,1).*(dat.X(:,1)*beta(1)+dat.X(:,2)*beta(2)+ ...
dat.X(:,3)*beta(3)+dat.X(:,4)*beta(4)+dat.X(:,5)*beta(5) ...
+dat.X(:,6)*beta(6)))/numy;
beta = a(i)*ones(6,1);
tic
beta1(i,:)=fminsearch(fun1,beta);
Tm(1,i)=toc;

% Second Approach

beta =a(i)*ones(6,1);
```

```
fVal=loglike(dat.y,dat.X,beta');
iJac=inv(myJac('loglike',dat.y,dat.X,beta'));
maxit = 100;
tol = 1e-6;
tic
for iter = 1:maxit
fnorm = norm(fVal);
if norm(fVal) < tol
break
end
d = - (iJac * fVal);
beta = beta+d;
fOld = fVal;
fVal = loglike(dat.y,dat.X,beta');
u = iJac*(fVal - fOld);
iJac = iJac + ( (d - u) * (d'*iJac) )/ (d'*u);
end
Tm(2,i)=toc;
beta2(i,:)=beta;

% Third Approach

fun3=@(beta)(-exp(dat.X(:,1)*beta(1)+dat.X(:,2)*beta(2)+ ...
dat.X(:,3)*beta(3)+dat.X(:,4)*beta(4)+dat.X(:,5)*beta(5) ...
+dat.X(:,6)*beta(6))+dat.y(:,1))/numy;
beta =  a(i)*ones(6,1);
tic
beta3(i,:)=lsqnonlin(fun3,beta);
Tm(3,i)=toc;

% Fourth Approach

fun4=@(beta)sum((-exp(dat.X(:,1)*beta(1)+dat.X(:,2)*beta(2)+ ...
dat.X(:,3)*beta(3)+dat.X(:,4)*beta(4)+dat.X(:,5)*beta(5) ...
+dat.X(:,6)*beta(6))+dat.y(:,1)).^2)/numy;
beta =a(i)*ones(6,1);
tic
beta4(i,:)=fminsearch(fun4,beta);
Tm(4,i)=toc;

end
```

```
% Time
Tm =

0.2949    0.5163    0.4666    0.3180    0.5153    0.4753
0.0196    0.0739    0.0765    0.0755    0.0738    0.0735
0.0382    0.0754    0.1221    0.1633    0.2047    0.2540
0.3233    0.2870    0.3521    0.3190    0.3695    0.3679

% First Approach
beta1 =

0.1688    0.0047    0.1010   -0.2556    0.1471   -0.2468
2.5346   -0.0323    0.1157   -0.3541    0.0798   -0.4095
2.5339   -0.0323    0.1157   -0.3540    0.0798   -0.4094
1.5489   -0.0152    0.1085   -0.3118    0.1064   -0.3499
2.5348   -0.0323    0.1157   -0.3539    0.0799   -0.4095
2.5338   -0.0323    0.1157   -0.3540    0.0799   -0.4095

% Second Approach
beta2 =

2.5339   -0.0323    0.1157   -0.3540    0.0798   -0.4094
NaN       NaN       NaN       NaN       NaN       NaN
-9.7293    0.2112    0.2275    0.2115    0.2110    0.2112
-16.3059   0.3100    0.3098    0.3100    0.3100    0.3100
-23.3383   0.4100    0.4100    0.4100    0.4100    0.4100
-30.5392   0.5100    0.5099    0.5100    0.5100    0.5100

% Third Approach
beta3 =

2.5128   -0.0384    0.1141   -0.2796    0.0676   -0.3695
2.5102   -0.0383    0.1140   -0.2808    0.0680   -0.3695
2.5099   -0.0383    0.1140   -0.2805    0.0678   -0.3690
2.5080   -0.0382    0.1138   -0.2805    0.0681   -0.3698
2.5078   -0.0381    0.1138   -0.2807    0.0682   -0.3701
2.5123   -0.0384    0.1141   -0.2799    0.0677   -0.3694

% Fourth Approach
beta4 =

2.5125   -0.0384    0.1141   -0.2796    0.0677   -0.3698
-0.8468  -0.0290    0.1630   -0.0279    0.2793   -0.2356
2.4753   -0.0387    0.1160   -0.2777    0.0694   -0.3716
1.7553   -0.0341    0.0871    0.1301    0.0539   -0.5094
2.9462   -0.0474    0.1165   -0.3151    0.0545   -0.3664
0.6875   -0.0153    0.1218   -0.2568    0.1270   -0.2133
```