Econ 512 HW 3

Kalyani Padmakumar

October 2018

The codes have been specified in the MATLAB file hw3˙updatedcodes.m.Results are as follows:

1 Question 1: Result

The estimated beta vector is denoted by sol1˙base here. More details are displayed along with the time to convergence:

```
sol1_base =

    2.5339
   -0.0323
    0.1157
   -0.3540
    0.0798
   -0.4094


fval =

   1.4270e+03


exitflag =

     1


output =

  struct with fields:

    iterations: 909
     funcCount: 1442
     algorithm: 'Nelder-Mead simplex direct search'
       message: 'Optimization terminated:↵ the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04 ↵ and F(X) satisfies the convergence criteria

Elapsed time is 0.052605 seconds.
>>
```

2 Question 2: Result

The estimated beta vector is denoted by sol2˙base here. More details are displayed along with the time to convergence:

```
sol2_base =

    0.9825
         0
    0.7368
    0.9196
    0.8973
    0.9142


fval2 =

   7.1555e+12


exitflag2 =

     1


output2 =

  struct with fields:

       iterations: 1
        funcCount: 14
         stepsize: 1.0459
     lssteplength: 1.0640e-41
    firstorderopt: 3.1287e+14
        algorithm: 'quasi-newton'
          message: 'Local minimum found.↵↵Optimization completed because the size of the gradient is less than↵the selected value of the optimality tolerance.↵↵Stopping 

Elapsed time is 0.009617 seconds.
>> 
```

3 Question 3: Result

The estimated beta vector is denoted by sol3˙base here. More details are displayed along with the time to convergence:

```
sol3_base =

  -22.9709
    1.0000
    0.9981
    1.0000
    1.0000
    1.0000

Elapsed time is 0.019607 seconds.
>>
```

4 Question 4: Result

The estimated beta vector is denoted by sol4˙base here. More details are displayed along with the time to convergence:

```
sol4_base =

    3.7477
   -0.0723
    0.1281
   -0.2249
    0.0422
   -0.4672


fval4 =

   5.6658e+03


exitflag4 =

     0


output4 =

  struct with fields:

    iterations: 1200
     funcCount: 1887
     algorithm: 'Nelder-Mead simplex direct search'
       message: 'Exiting: Maximum number of iterations has been exceeded↵        - increase MaxIter option.↵        Current function value: 5665.800229 ↵'

Elapsed time is 0.029813 seconds.
>>
```

5 Question 5: Result

In terms of time to convergence, the quasi-Newton method converges to the solution quickly, followed by lsqnonlin in question 3, then the nelder mead simplex method of estimation by NLS and then the nelder mead simplex method of estimation by MLE.

In order to check robustness, we have recalculated solutions for questions 1-4 for 10 sets of initial values for these parameters. Each set of initial value is denoted by beta0, where each beta0 is a column vector in r. r is a (10*6) matrix of numbers uniformly distributed between 0 and 10.

For each vector of initial values, I have calculated the norm of the difference between the new solution and the original solution obtained in questions 1/2/3/4. The results have been described below.

The vector x displays the norm of the difference between the new solution for each vector of initial values and the original solution obtained in question 1. For instance, the element in the first row and first column of x displays the norm of the difference between the solution (when the initial vector is the first column of r) and the solution obtained in question 1.Similarly, the vector w displays the norm of the difference between the new solution for each vector of initial values and the original solution obtained in question 2.Vector z displays the norm of the difference between the new solution for each vector of initial values and the original solution obtained in question 3.Vector a displays the norm of the difference between the new solution for each vector of initial values and the original solution obtained in question 4.

From below, it looks as though the Quasi Newton method is least sensitive to the initial condition.

```
>> x

x =

   19.9119   18.4889    0.0214   21.7494   13.9297    6.2358  103.9806   50.2844   56.9082    9.2550

>> w

w =

   11.1695   12.3861   12.9524   11.7805   10.5775   10.6273   15.0340    9.4687   12.8715   12.7597

>> z

z =

   28.0022   29.1740   27.4122   29.9991   28.8116   14.7421   34.4744   31.0524   29.2188   31.9350

>> a

a =

   24.3000   13.2364   14.4491   12.3916   11.9283  269.2220   15.4640   48.7763   50.7381   67.9637

>>
```

(Please note that this is joint work with Ece Teoman and we referred to the web for help with certain commands.)