Homework 3: ECON512

Joonkyo Hong

1. $\sim$ 4.

For all methods, I start off with the following initial value:

$$\tilde{\beta} = (X'X)^{-1}X'\log(1+y)$$

If the model specification is correct, $\tilde{\beta}$ is incorrect because $\log E(y|x) \neq E(\log y|x)$. However, it provides somewhat reasonable an initial value for MLE and NLS.

I perform four approaches:

1) MLE – NM: Estimating the parameter β via the method of maximum likelihood with Nelder-Mead algorithm (`fminsearch` is used).

2) MLE – QN: Estimating the parameter β via the method of maximum likelihood with quasi-Newton algorithm (`fminunc` with `quasi-newton` algorithm is used).

3) NLS – LQ: Estimating the parameter β via the method of nonlinear least squares with `lsqnonlin`.

4) NLS – NM: Estimating the parameter β via the method of nonlinear least squares with Nelder-Mead algorithm (`fminsearch` is used).

Table 1: Parameter Estimates

Dep. var:# of affairs	MLE–NM	MLE–QN	NLS–LQ	NLS–NM
Age	-0.0322	-0.0322	-0.0383	-0.0384
Married Year	0.1157	0.1157	0.1141	0.1141
Religiousness	-0.3540	-0.3540	-0.2797	-0.2796
Occupation	0.0798	0.0798	0.0677	0.0676
Self-rating marriage	-0.4094	-0.4094	-0.3699	-0.3698
Const.	2.5339	2.5339	2.5121	2.5126

5.

In order to explore how does each approach respond to different initial values, I estimate parameters via the four approaches with 187s' many initial values. In particular, I create 17 candidates for a initial value for β_0 and 11 candidates for a initial value for β_6, and let all the other values remain unchanged from $\tilde{\beta}$. So in this exercise, β_0 spans from zero to four with grid 0.25, indexed by m and β_6 spans from -0.5 to zero with grid 0.05, indexed by n. For instance, the (4,5)th initial value is going to be

$$(0.75, -0.01327, 0.04034, -0.1204, 0.0284, -0.3)',$$

and the (7,1)th initial value is going to be

$$(1.5, -0.01327, 0.04034, -0.1204, 0.0284, -0.5)'.$$

Then, I measure the robustness of each approach to initial choice by the following formula:

$$R_j = (\frac{1}{6}\frac{1}{N}\frac{1}{M}\sum_m\sum_n\sum_k(\hat{\beta}_k^j(m,n) - \hat{\beta}_k^j)^2)^{-1},$$

where j refers to an approach, $\hat{\beta}^j$ is a vector of the estimates through j-approach, which are reported in Table 1, and $\hat{\beta}^j(m,n)$ is a vector of the estimates thorough j-approach with (m,n)-th initial value. It immediately follows that the larger R_j is, the more robust the j approach is.

Table 2 reports the performance measures for each approach. Notably, Nelder-Mead method does a good job at estimating parameters whatever an initial value is. In particular, the corresponding R_j values to MLE-NM and NLS-NM are 145.91 and 88.76, respectively. In contrast, quasi-Newton or `lsqnonlin` algorithm is highly responsive to a initial value. NLS-LQ is the worst one. When it comes to Time to convergence, there is no clear winner among the four approaches. Albeit fastest, NLS-LQ is not remarkably faster than all the other approaches. Overall, MLE-NM approach is the best one as its robustness measure is largest.

Table 2: Performance

Performance Measure	MLE–NM	MLE–QN	NLS–LQ	NLS–NM
$R_j \times$ 10e-10	145.91	1.06	0.025	88.76
Time to Convergence	0.023	0.033	0.013	0.024

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Homework #3 ECON 512                                     %
% Written by Joonkyo (Jay) Hong, 4 Oct 2018                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;

% Load the data
load hw3.mat

% Set the options
options_nm = optimset('Display','off','MaxFunEvals',5000,'MaxIter',5000,'TolFun',10e-6,'TolX',10e-6);
options_qn = optimoptions(@fminunc,'Display','off','Algorithm','quasi-newton');
options_nls = optimoptions(@lsqnonlin,'Display','off','Algorithm','levenberg-marquardt',...
'MaxFunctionEvaluations',10000,'MaxIterations',5000,'TolX',10e-8);

% Initial Value: OLS estimate for log(y+1) = X*beta + u
% Of course this is bad specifiction when the DGP is true because
% E(log(y)|x) =/= log E(y|x)

% But it provides us with somewhat "reasonable" initial value for
% MLE,NLS,GMM...

init_theta = (X'*X)\X'*log(1+y);

%% Questions 1 and 2 (MLE)

% Negative Likelihood Function
lik_fcn = @(theta) -sum(-exp(X*theta)+y.*(X*theta));

% Nelder-Mead
tic
[thetamle_nm,~,~,~] = fminsearch(lik_fcn,init_theta,options_nm);
t_mlenm = toc;

% Quasi-Newton
tic
[thetamle_qn,~,~,~] = fminunc(lik_fcn,init_theta,options_qn);
t_mleqn = toc;

%% Questions 3 and 4 (NLS)

% Nonlinear Least Square objective function
obj_fcn_lsq = @(theta) (y-exp(X*theta));
```

```
% Least Square Non-linear
tic
thetanls_lq = lsqnonlin(obj_fcn_lsq,init_theta,[],[],options_nls);
t_nlslq = toc;

% Nelder-Mead
obj_fcn = @(theta) (y-exp(X*theta))'*(y-exp(X*theta));
tic
[thetanls_nm,~,~,~] = fminsearch(obj_fcn,init_theta,options_nm);
t_nlsnm = toc;

%% Question 5 (Check the Robustness to Initial Value)

initbeta0 = 0:0.25:4;
initbeta6 = -0.5:0.05:0;

M = length(initbeta0);
N = length(initbeta6);
mle_nm = zeros(M,N);
mle_qn = zeros(M,N);
nls_lq = zeros(M,N);
nls_nm = zeros(M,N);

    for m=1:M
        for n=1:N

            init_theta = [initbeta0(m);init_theta(2:5,1);initbeta6(n)];

            x = fminsearch(lik_fcn,init_theta,options_nm);
            mle_nm(m,n) = (x-thetamle_nm)'*(x-thetamle_nm)/6;

            x = fminunc(lik_fcn,init_theta,options_qn);
            mle_qn(m,n) = (x-thetamle_qn)'*(x-thetamle_qn)/6;

            x = lsqnonlin(obj_fcn_lsq,init_theta,[],[],options_nls);
            nls_lq(m,n) = (x-thetanls_lq)'*(x-thetanls_lq)/6;

            x = fminsearch(obj_fcn,init_theta,options_nm);
            nls_nm(m,n) = (x-thetanls_nm)'*(x-thetanls_nm)/6;

        end
     end
```

```
robustness_measure = 10e-10*(1./[mean(mean(mle_nm)) mean(mean(mle_qn))...
mean(mean(nls_lq)) mean(mean(nls_nm))]);
convergence_time = [t_mlenm t_mleqn t_nlslq t_nlsnm];

 disp("              ");
 disp("RESULT");
 disp("Parameter estimates of each method");
 disp("MLE_NM        MLE_QN        NLS_LQ      NLS_NM");
 disp(num2str([thetamle_nm thetamle_qn thetanls_lq thetanls_nm]));
  disp("              ");
 disp("Robustness Measure of each method");
 disp("MLE_NM        MLE_QN        NLS_LQ      NLS_NM");
 disp(num2str(robustness_measure));
 disp("              ");
 disp("Time to Convergence of each method");
 disp("MLE_NM        MLE_QN        NLS_LQ      NLS_NM");
 disp(num2str(convergence_time));
```