Penn State, Fall 2018, ECON512 Empirical Method

Homework 3

Kensuke Suzuki

October 4, 2018

Problem 1

First define the function `TobitLLF.m` which gives the (negative of) log likelihood for given vector of β, $\mathbf{X}$ and $\mathbf{y}$.

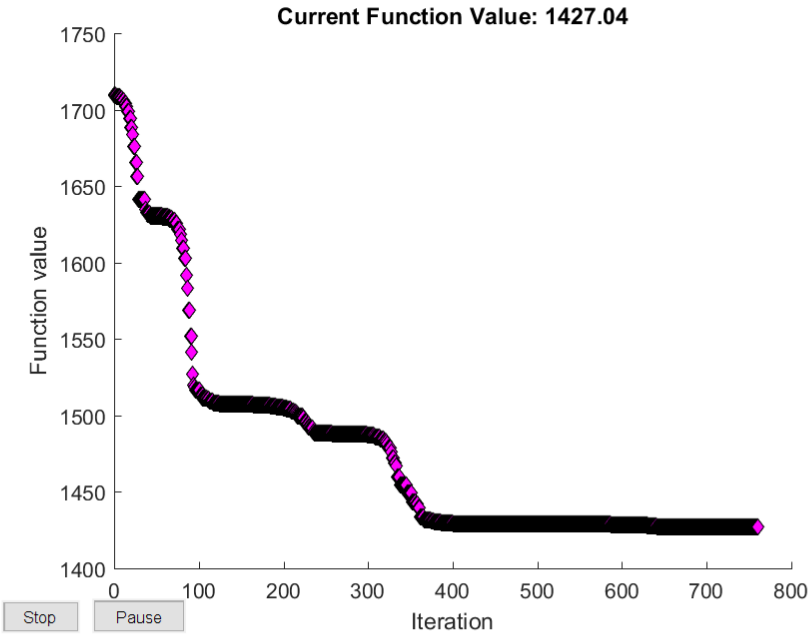
```
function LLF = TobitLLF(beta,X,y)
f = -exp(X*beta) + y .* (X*beta) - log(factorial(y)) ;
%f = -exp(X*beta) + y .* (X*beta) ;
LLF = - sum(f);
end
```

Given the vector of $\mathbf{X}$ and $\mathbf{y}$, we re-define anonymous function of β which we will minimize with respect to the vector of β.
In the first question, we use the Nelder-Mead method. Following the course material, we use the build-in optimizer `fminsearch`.

For the initial value, we use $\beta_0 = \left[\log \frac{\sum_i y_i}{n}, 0, 0, 0, 0, 0\right]^\top$. Optimization algorithm yields the MLE estimates:

$$\beta = \begin{bmatrix} 2.5339 \\ -0.0323 \\ 0.1157 \\ -0.3540 \\ 0.0798 \\ -0.4094 \end{bmatrix}$$

The figure below shows the function values along the iteration.



Problem 2

In this question, we use the build-in optimizer `fminunc`, where the default algorithm is quasi-Newton method. The algorithm found the local minimum as the size of the gradient is less than the tolerance. Estimated MLE estimates are:

$$\beta = \begin{bmatrix} 2.5339 \\ -0.0323 \\ 0.1157 \\ -0.3540 \\ 0.0798 \\ -0.4094 \end{bmatrix}$$

The estimates are same as in the question 1.

Problem 3

We now define the function returning the (negative of) sum of squared residuals, `NlsRSS`.

```
function RSS = NlsRSS(beta,X,y)
res = y - exp(X*beta);
RSS = - (res' * res);
end
```

Given vector of $\vec{X}$ and $\mathbf{y}$, we define the anonymous function of β which we minimize.
We use the same initial value for β. As we discussed in the last question, the estimates are greatly affected by the initial guess.
For this question, we use the built-in optimizer `lsqnonlin`. The results are:

$$\beta = \begin{bmatrix} 0.3755 \\ -0.0119 \\ 0.0856 \\ -0.0503 \\ 0.0187 \\ -0.0745 \end{bmatrix}$$

Problem 4

In this question, we employ the Nelder-Mead method using `fminsearch`. Initial value for β is same as before. Again, as we discussed below, the results are greatly affected by the initial values. The optimization yields:

$$\beta = \begin{bmatrix} 345.9665 \\ 1.2039 \\ -7.8108 \\ 1.8421 \\ -0.5917 \\ 0.9241 \end{bmatrix}$$

Problem 5

For each method, we minimizes the function value with different initial values for β and see how the results are affected by the initial values. For here, we change the initial values for β_0 from 1 to 5 (with step 0.2) and keep $\beta_i = 0$ for $i = 1, ..., 5$

MLE with Nelder-Mead (Figure 1)

Figure 1 shows the estimated coefficients with different initial values for β_0. It is clear from the figure that the estimates vary substantially with different initial values. The response of estimates seems to be random to the initial values.

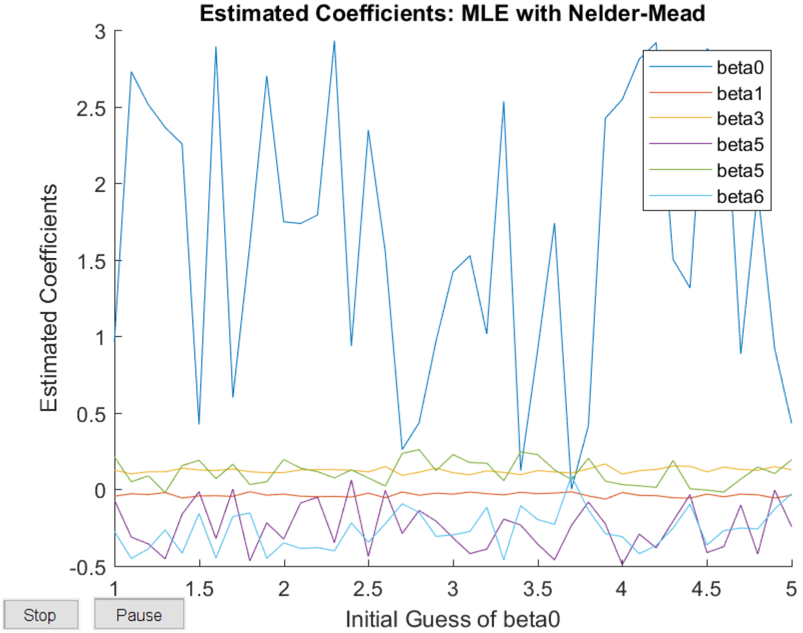
MLE with Quasi-Newton (Figure 2)

Figure 2 is the result with Quasi-Newton method. Unlike the Nelder-Mead method, the Quasi-Newton method yields fairly stable results. It implies that the results are not substantially affected by the initial values.

NLS with `lsqnonlin` (Figure 3)

Again the results are greatly affected by the initial values but the figure reveals (unlike the case in the Nelder-Mead presented below) linear relationship between the estimates and initial values.

Figure 1: MLE with Nelder-Mead



NLS with Nelder-Mead (Figure 4)

Finally, the figure 4 demonstrates the results for NLS with Nelder-Mead. The left y-axis is the coefficients for β_0 and the right y-axis is for the rest of the parameters. As in Figure 1, it reveals that the initial values matter in the Nelder-Mead method. We find (in particular for β_0) mostly random fluctuation in the results with different initial values.

Discussion:

Matlab Code

```
% ECON512 Homework 3
% Kensuke Suzuki
clear all
delete HW3log.txt
diary('HW3log.txt')
diary on

disp('ECON512 HOMEWORK3: Ken Suzuki')

disp(' ')

data = load('hw3.mat');
X = data.X;
y = data.y;
Parameters = {'beta0'; 'beta1'; 'beta2'; 'beta3';'beta4'; 'beta5'};
```

Figure 2: MLE with Quasi-Newton

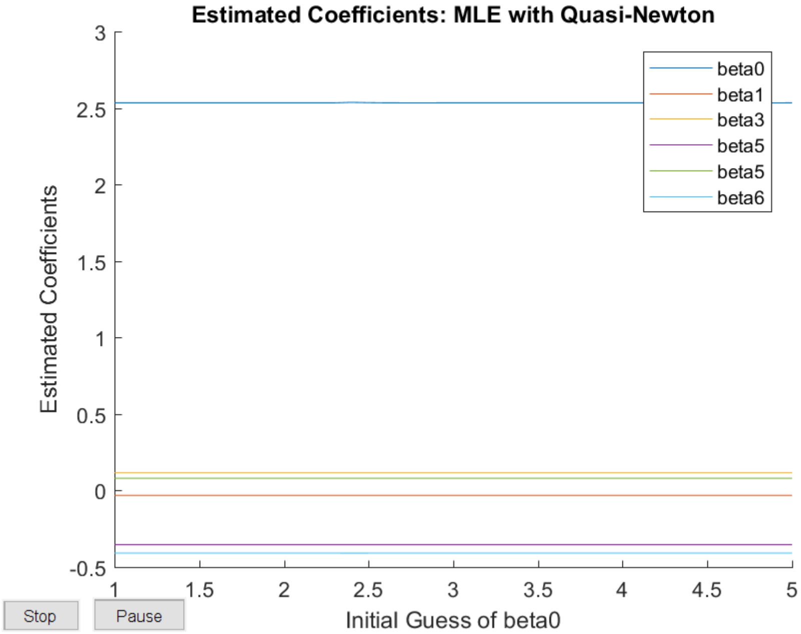


Figure 3: NLS with lsqnonlin

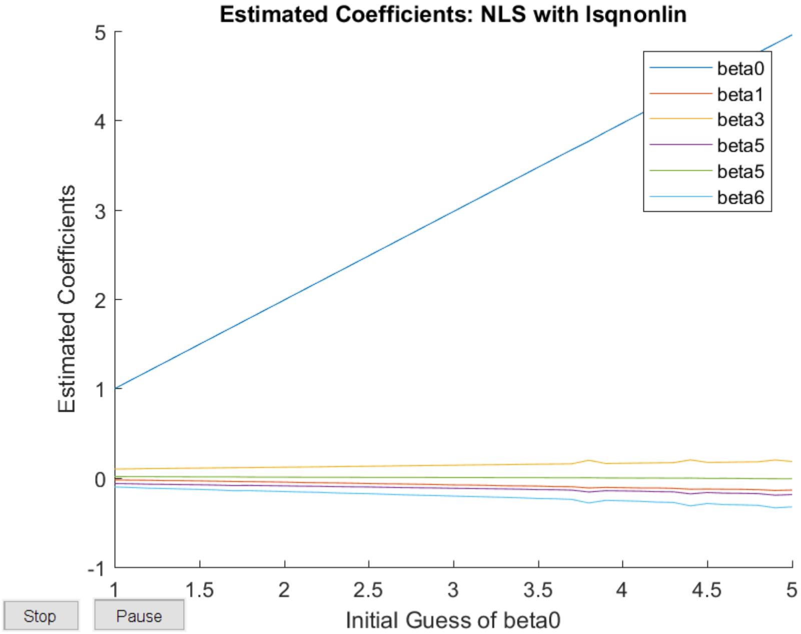
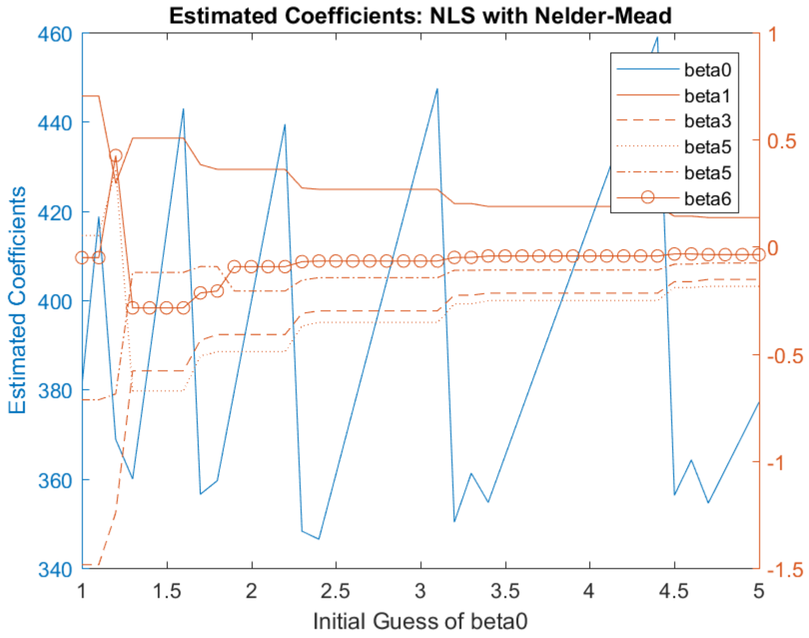


Figure 4: NLS with Nelder-Mead



```
%bindcell = cellstr(bind);
%% Problem 1: Nelder Mead Simplex Method

options = optimset('PlotFcns',@optimplotfval, 'Display','iter');

TobitLLF_beta =  @(beta) TobitLLF(beta,X,y);
%beta0 = [1,1,1,1,1,1]';
beta0 = [log(mean(y)),zeros(1,5)]';


beta_p1 = fminsearch(TobitLLF_beta, beta0, options);
saveas(gcf, 'q1.png')
%beta_p1 = fminsearch(TobitLLF_beta, beta0);

EstimatedCoeff = beta_p1;
LLF= −TobitLLF(beta_p1,X,y)
disp('———Problem 1———')
disp('Estimated Parameter Vector')
Result_Q1 = table(Parameters, EstimatedCoeff)

%% Problem 2: Quasi−Newton Optimization Method
clear beta
beta_p2 = fminunc(TobitLLF_beta, beta0)
EstimatedCoeff = beta_p2;

```

```
disp('———Problem 2———')
disp('Method: Quasi–Newton Optimization (fminunc)')
disp('Estimated Parameter Vector')
Result_Q2 = table(Parameters, EstimatedCoeff)


%% Problem 3:

NlsRSS_beta =  @(beta) NlsRSS(beta,X,y);

beta_p3_1 = lsqnonlin(NlsRSS_beta,beta0)
beta_p3_2 = lsqnonlin(NlsRSS_beta,beta_p1)

Coeff1 = beta_p3_1;
Coeff2 = beta_p3_2;

%% Problem 3: NLS with lsqnonlin

NlsRSS_beta =  @(beta) NlsRSS(beta,X,y);

beta_p3_1 = lsqnonlin(NlsRSS_beta,beta0)
beta_p3_2 = lsqnonlin(NlsRSS_beta,beta_p1)
Coeff1 = beta_p3_1;
Coeff2 = beta_p3_2;

disp('———Problem 3———')
disp('NLS with lsqnonlin')
disp('Estimated Parameter Vector')
disp('Coeff1: [log(mean(y)),zeros(1,5)] as initial guess')
disp('Coeff2: MLE estimator (question 1) as initial guess')
Result_Q3 = table(Parameters, Coeff1, Coeff2)


%% Problem 4: NLS with Nelder–Mead

beta_p4_1 = fminsearch(NlsRSS_beta,beta0)
beta_p4_2 = fminsearch(NlsRSS_beta,beta_p1)
Coeff1 = beta_p4_1;
Coeff2 = beta_p4_2;

disp('———Problem 4———')
disp('NLS with Nelder–Mead')
disp('Estimated Parameter Vector')
disp('Coeff1: [log(mean(y)),zeros(1,5)] as initial guess')
disp('Coeff2: MLE estimator (question 1) as initial guess')
Result_Q3 = table(Parameters, Coeff1, Coeff2)

%% Problem 5
```

```

intbeta0 = [1:0.1:5];

% MLE with fminsearch
MLE_fmins = zeros(6,size(intbeta0,2));
for i = 1:size(intbeta0,2)
    beta0 = [intbeta0(1,i),zeros(1,5)]';
    beta = fminsearch(TobitLLF_beta, beta0);
    MLE_fmins(:,i)=beta;
end
result_MLE_fmins = [intbeta0; MLE_fmins];

%MLE with quasi newton
MLE_fminunc = zeros(6,size(intbeta0,2));
for i = 1:size(intbeta0,2)
    beta0 = [intbeta0(1,i),zeros(1,5)]';
    beta = fminunc(TobitLLF_beta, beta0);
    MLE_fminunc(:,i)=beta;
end
result_MLE_fminunc = [intbeta0; MLE_fminunc];

%NLS with lsqnonlin
NLS_lsqnonlin = zeros(6,size(intbeta0,2));
for i = 1:size(intbeta0,2)
    beta0 = [intbeta0(1,i),zeros(1,5)]';
    beta = lsqnonlin(NlsRSS_beta,beta0);
    NLS_lsqnonlin(:,i)=beta;
end
result_NLS_lsqnonlin = [intbeta0; NLS_lsqnonlin];

%NLS with lsqnonlin
NLS_fmins = zeros(6,size(intbeta0,2));
for i = 1:size(intbeta0,2)
    beta0 = [intbeta0(1,i),zeros(1,5)]';
    beta = fminsearch(NlsRSS_beta,beta0);
    NLS_fmins(:,i)=beta;
end
result_NLS_fmins = [intbeta0; NLS_fmins];

% MLE with fminsearch
plot(intbeta0, result_MLE_fmins(2,:), ...
    intbeta0, result_MLE_fmins(3,:), ...
    intbeta0, result_MLE_fmins(4,:), ...
    intbeta0, result_MLE_fmins(5,:), ...
    intbeta0, result_MLE_fmins(6,:), ...
    intbeta0, result_MLE_fmins(7,:))
title('Estimated Coefficients: MLE with Nelder-Mead')
xlabel('Initial Guess of beta0')
```

```
ylabel('Estimated Coefficients')
legend('beta0','beta1', 'beta3', 'beta5', 'beta5', 'beta6')
saveas(gcf, 'MLE_NM.png')

% MLE with Quasi Newton
plot(intbeta0, result_MLE_fminunc(2,:), ...
    intbeta0, result_MLE_fminunc(3,:), ...
    intbeta0, result_MLE_fminunc(4,:), ...
    intbeta0, result_MLE_fminunc(5,:), ...
    intbeta0, result_MLE_fminunc(6,:), ...
    intbeta0, result_MLE_fminunc(7,:))
title('Estimated Coefficients: MLE with Quasi-Newton')
xlabel('Initial Guess of beta0')
ylabel('Estimated Coefficients')
legend('beta0','beta1', 'beta3', 'beta5', 'beta5', 'beta6')
saveas(gcf, 'MLE_QN.png')

% NLS with lsqnonlin
plot(intbeta0, result_NLS_lsqnonlin(2,:), ...
    intbeta0, result_NLS_lsqnonlin(3,:), ...
    intbeta0, result_NLS_lsqnonlin(4,:), ...
    intbeta0, result_NLS_lsqnonlin(5,:), ...
    intbeta0, result_NLS_lsqnonlin(6,:), ...
    intbeta0, result_NLS_lsqnonlin(7,:))
title('Estimated Coefficients: NLS with lsqnonlin')
xlabel('Initial Guess of beta0')
ylabel('Estimated Coefficients')
legend('beta0','beta1', 'beta3', 'beta5', 'beta5', 'beta6')
saveas(gcf, 'NLS_lsqnonlin.png')

% NLS with fminsearch
figure
yyaxis left
plot(intbeta0, result_NLS_fmins(2,:))
xlabel('Initial Guess of beta0')
ylabel('Estimated Coefficients')
title('Estimated Coefficients: NLS with Nelder-Mead')
yyaxis right
plot(intbeta0, result_NLS_fmins(3,:), ...
    intbeta0, result_NLS_fmins(4,:), ...
    intbeta0, result_NLS_fmins(5,:), ...
    intbeta0, result_NLS_fmins(6,:), ...
    intbeta0, result_NLS_fmins(7,:))
legend('beta0','beta1', 'beta3', 'beta5', 'beta5', 'beta6')
saveas(gcf, 'NLS_NM.png')

diary off
```