IO Assignment Part 2

Zikai Xu

November 29, 2023

1. (a) In Ω, there are 10 regions, 5 gravity states, 20 DAV grids, 3 violator status and 4 possible lagged investment. Let Ω_1 denotes the fixed states of region and gravity, we find that there are 45 states that Ω_1 can take.

   If the plant is in compliance, its DAV and two quarterly lags of investment must be 0. So there are 45 states if the plant is in compliance and $45 \times 20 \times 2 \times 4 = 7200$ states if the plant is not in compliance. There are $7,245$ states in Ω.

   (b) In $\tilde{\Omega}$, there are 10 regions, 5 gravity states, 20 DAV grids, 3 violator status, 2 possible lag investment and 2 violations. Similarly there are 45 states in Ω_1 and when the plant is in compliance, its DAV, lag investment and violation must be 0. Hence, there are also 7245 states in $\tilde{\Omega}$.

   (c) The investment probability for all states with $DAV = 2$ is as follows:

```
(:,:,2,1,1) =

    0.0991
    0.0983
    0.0995
    0.1065
    0.1043
    0.1055
    0.1232
    0.1128
    0.1198
    0.1218
    0.1246
    0.1095
    0.1048
    0.1090
    0.1077
    0.1101
    0.1066
    0.0984
    0.1073
    0.1045
    0.1058
    0.1230
    0.1078
```

```
    0.1217
    0.1215
    0.1238
    0.1111
    0.1050
    0.1109
    0.1101
    0.1103
    0.0930
    0.0859
    0.0916
    0.0918
    0.0926
    0.1293
    0.1082
    0.1276
    0.1230
    0.1315
    0.1191
    0.1033
    0.1188
    0.1248


(:,:,3,1,1) =

    0.3419
    0.3112
    0.3422
    0.4259
    0.3910
    0.4292
    0.4359
    0.3998
    0.4292
    0.4033
    0.4514
    0.3926
    0.3748
    0.3911
    0.3585
    0.3986
    0.3307
    0.3119
    0.3337
    0.2973
```

```
    0.3323
    0.3677
    0.3549
    0.3705
    0.3321
    0.3670
    0.3336
    0.3312
    0.3399
    0.3103
    0.3345
    0.3849
    0.3656
    0.3858
    0.3499
    0.3842
    0.4059
    0.3477
    0.3851
    0.3476
    0.4117
    0.3952
    0.3560
    0.3547
    0.4051


(:,:,2,2,1) =

    0.0939
    0.0918
    0.0937
    0.1069
    0.1038
    0.1052
    0.1248
    0.1146
    0.1213
    0.1239
    0.1264
    0.1122
    0.1084
    0.1121
    0.1105
    0.1125
```

```
    0.1079
    0.0979
    0.1090
    0.1047
    0.1064
    0.1287
    0.1093
    0.1261
    0.1270
    0.1298
    0.1126
    0.1068
    0.1125
    0.1114
    0.1115
    0.0818
    0.0704
    0.0806
    0.0788
    0.0803
    0.1500
    0.1148
    0.1454
    0.1389
    0.1529
    0.1220
    0.1060
    0.1217
    0.1276


(:,:,3,2,1) =

    0.3001
    0.2783
    0.2989
    0.3486
    0.3253
    0.3496
    0.3146
    0.2944
    0.3108
    0.2946
    0.3164
    0.3226
    0.3162
```

```
    0.3225
    0.2991
    0.3251
    0.2498
    0.2444
    0.2494
    0.2316
    0.2494
    0.2644
    0.2662
    0.2654
    0.2455
    0.2616
    0.2607
    0.2673
    0.2658
    0.2463
    0.2596
    0.2586
    0.2533
    0.2564
    0.2447
    0.2559
    0.3075
    0.2857
    0.3025
    0.2738
    0.3125
    0.3345
    0.3090
    0.3046
    0.3394

(:,:,2,1,2) =

    0.1206
    0.1198
    0.1223
    0.1324
    0.1321
    0.1337
    0.1533
    0.1323
    0.1462
    0.1556
    0.1607
```

```
    0.1287
    0.1168
    0.1260
    0.1284
    0.1313
    0.1346
    0.1144
    0.1350
    0.1320
    0.1360
    0.1709
    0.1334
    0.1666
    0.1690
    0.1771
    0.1325
    0.1195
    0.1305
    0.1338
    0.1337
    0.1019
    0.0909
    0.1020
    0.0994
    0.1005
    0.1675
    0.1378
    0.1666
    0.1538
    0.1641
    0.1721
    0.1285
    0.1747
    0.1882


(:,:,3,1,2) =

    0.4291
    0.3795
    0.4332
    0.5413
    0.4960
    0.5541
    0.5122
    0.4483
```

```
    0.4987
    0.4679
    0.5296
    0.4979
    0.4516
    0.4908
    0.4534
    0.5091
    0.4229
    0.3773
    0.4213
    0.3700
    0.4292
    0.4388
    0.4086
    0.4386
    0.3848
    0.4383
    0.3913
    0.3770
    0.3947
    0.3584
    0.3934
    0.4530
    0.4117
    0.4492
    0.4018
    0.4544
    0.5747
    0.4436
    0.5377
    0.4840
    0.5825
    0.5459
    0.4505
    0.4840
    0.5655

(:,:,2,2,2) =

    0.1316
    0.1303
    0.1346
    0.1391
    0.1399
    0.1415
```

```
    0.1516
    0.1314
    0.1440
    0.1551
    0.1595
    0.1284
    0.1181
    0.1260
    0.1285
    0.1304
    0.1429
    0.1174
    0.1410
    0.1404
    0.1453
    0.1843
    0.1376
    0.1740
    0.1843
    0.1931
    0.1332
    0.1204
    0.1304
    0.1352
    0.1347
    0.1031
    0.0829
    0.1037
    0.0978
    0.1010
    0.2116
    0.1559
    0.2022
    0.1881
    0.2061
    0.1704
    0.1294
    0.1745
    0.1849


(:,:,3,2,2) =

    0.3448
    0.3098
    0.3425
```

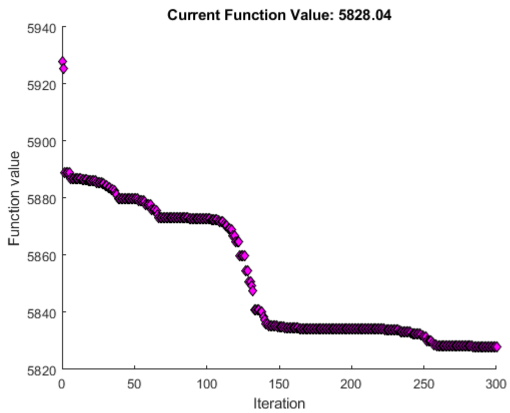
```
0.4057
0.3759
0.4082
0.3498
0.3048
0.3412
0.3235
0.3483
0.3771
0.3558
0.3732
0.3478
0.3786
0.2855
0.2672
0.2805
0.2590
0.2834
0.2865
0.2787
0.2845
0.2596
0.2794
0.2830
0.2808
0.2848
0.2633
0.2795
0.2751
0.2585
0.2699
0.2552
0.2699
0.3573
0.3271
0.3549
0.3164
0.3622
0.4234
0.3692
0.3779
0.4292
```

When $DAV = 2$, the status must be either regular violator or HPV. So there are 8 columns. And each row in a column represents a Ω_1 state.

2. (a) The quasi-likelihood for the parameter vector $\hat{\theta}$ is

```
            The log likelihood of theta hat is:
-6.1847e+03
```

(b) The $-loglikelihood$ varies according to iterations as follows:



The maximum log likelihood I can find is -5828.04. The corresponding θ^{ML} is $(2.8451, -1.0713, -1.7442, -3.9443, 0.3102)$.

(c) The standard errors are summarized in the following variance covariance matrix.

```
          >> disp(VarML);
  1.0e+03 *

  -0.9131    1.2188   -1.1067   -1.2077    2.0087
   1.2188    1.1349    2.8605   -2.5279   -2.6863
  -1.1067    2.8605    0.7286   -2.0181   -0.4643
  -1.2077   -2.5279   -2.0181    3.4128    2.3408
   2.0087   -2.6863   -0.4643    2.3408   -1.1990
```

3. (a) The implicit cost of investment is 2.8451 dollars. Implicit cost of HPV is -0.3102, cost of inspection is 1.0713, and cost of violation is 1.7442.

(b) The implicit cost of \$ 1 of fines is actually \$ 3.9443.

MATLAB code is attached:

```
    %% Read Data
clc
clear
% Table for Bellman computation
Bellman = readtable("data_for_bellman_computations.csv");
% Only take naics_recode value of 1
Bellman = Bellman((Bellman.naics_recode==1),:);
% Use Omega1 to represent the fixed states
Omega1 = Bellman.omega1; % It turns out there are 45 possible Omega1
% Table for Omega
T1 = readtable('value_part2_problem1_thetaBGL.csv');
Omega = T1{:,[1,5:8]};
Omega(:,1)=Omega1;
% Table for Omega tilde
T2 = readtable("valuetilde_part2_problem1_thetaBGL.csv");
Omegat = T2{:,[1,5:8]};
Omegat(:,1)=Omega(:,1);

%% 1) Computation of the plant's dynamic optimization decision for
% parameters

Coeff = [2,-0.5,-0.5,-5,-0.1]; %X,I,V,F,H
%Coeff = [2.872,-0.049,-0.077,-5.980,-0.065]; %BGL check
[NewV,Vtilde,Investprob]=Bellmanfun(Coeff,Omega,Omegat,Bellman);

disp(Investprob(:,4,:,:,:));

%% 2) Nested fixed point quasi-maximum likelihood estimation
% Read the sample
sample = readtable("analysis_data.csv");
% Only take naics_recode value of 1 and the plants are not comliant
sample = sample((sample.naics_recode==1 & sample.compliance==0),:);
somega = sample(:,{'omega1','DAV','ordered_violator','lag_investment','violation','investment'});

%% The quasi log-likelihood of parameter theta is
loglike = LogLike(Investprob,somega);
disp('The log likelihood of theta hat is:');
disp(loglike);

%% 2b Maximize the likelihood
% Starting value is thetaBGL
X0 = [2.872,-0.049,-0.077,-5.980,-0.065];
% Create a function such that Omega,Omegat,Bellman,somega are parameters.
% Matlab just needs to maximize likelihood by choosing theta.
```

```
fun = @(x)-loglikefun(x,Omega,Omegat,Bellman,somega);
option = optimset('Display','Iter','MaxIter',1000,'PlotFcns',@optimplotfval);
% Do at most 5 iterations in fminsearch function
[x,fval,exitflag,output] = fminsearch(fun,X0,option);
disp('Theta_ML')
disp(x);

%% 2c Find the standard errors
% First estimate the variance covariance matrix
VarML = varml(x,fval,Omega,Omegat,Bellman,somega);
% Standard errors are just the square root of diagnal of this matrix
disp(VarML);
```

Codes for functions:
Bellmanfun:

```
    function [NewV,Vtilde,Investprob] = Bellmanfun(Coeff,Omega,Omegat,Bellman)
tic
beta = 0.95^0.25;
gamma = 0;
N = size(Omega,1); % The number of states

% initialization
% Value function at the beginning of the period
NewV = zeros([45,20,3,2,2]); % Omega1, DAVgrid, violator status, lag_inv, lag2_inv
OldV = zeros([45,20,3,2,2]);
% Value function right after the regulator has moved.
Vtilde = zeros([45,20,3,2,2]); % Omega1, DAVgrid, violator status, lag_inv, violation
% Investment probability
Investprob = zeros([45,20,3,2,2]);% Omega1, DAVgrid, violator status, lag_inv, violation
% Regulation probability
P_regu = zeros(N,80);
% Transition probability
Tran = zeros(N,80,3);
% Static utility given a state Omega and a regulation action
U = zeros(N,80,3);
EU = zeros(N,80);
% Vtilde under each regulation action
Vr = zeros(N,2,3);
% Expected Vtilde conditional on inspection, violation and fines
EVr = zeros(N,80);
% The difference between new and old value function
norm = 1;

for j = 1:80
            % The probability of regulation action j is implemented is
```

```
            P_regu(:,j) = Bellman.(14+(j-1)*7);
            % Under regulation action j, the plant's violator status
            % becomes compliance, regular violator and HPV with probability
            % Tran0, Tran1 and Tran2 respectively.
            Tran(:,j,1) = Bellman.(15+(j-1)*7);
            Tran(:,j,2) = Bellman.(16+(j-1)*7);
            Tran(:,j,3) = Bellman.(17+(j-1)*7);
            % Static utility under Omega i and regulation action j and
            % transit to compliance.
            U(:,j,1) = Coeff(2)*Bellman.(11+(j-1)*7)+Coeff(3)*Bellman.(12+(j-1)*7)...
                + Coeff(4)*Bellman.(13+(j-1)*7);
            % transit to regular violator
            U(:,j,2) = Coeff(2)*Bellman.(11+(j-1)*7)+Coeff(3)*Bellman.(12+(j-1)*7)...
                + Coeff(4)*Bellman.(13+(j-1)*7);
            % transit to HPV
            U(:,j,3) = Coeff(2)*Bellman.(11+(j-1)*7)+Coeff(3)*Bellman.(12+(j-1)*7)...
                + Coeff(4)*Bellman.(13+(j-1)*7)+Coeff(5);
            % Expected utility conditional on Omega and inspection,
            % violation and fines
            EU(:,j) = dot(U(:,j,:),Tran(:,j,:),3);
 end
 % EEU is the expected static utility
 EEU = zeros(N,1);
 for i = 1:N
 EEU(i) = dot(EU(i,:),P_regu(i,:),2);
 end



%% Iteration
iter = 0;
while norm > 1e-6
    iter = iter + 1;
% Loop through states Omegatilde. For each Omegatilde
%% Upper bound and lower bound of OldV(Omega) next period. If
% X=1 in this period, then Omega in next
% period has Omega.lag_inv=1 and Omega.lag2_inv=Omegat.lag_inv
Vinv_u = zeros(N,1);
Vinv_l = zeros(N,1);
Vnotinv_u = zeros(N,1);
Vnotinv_l = zeros(N,1);
omega1 = Omegat(:,1); % Fixed state
DAV = Omegat(:,2); % From 0 to 9.5
status = Omegat(:,3); % Violation status
lag_inv = Omegat(:,4); % Lag 1 quarter investment
```

```
vio = Omegat(:,5); % Violation
DAVgrid_new = DAV*0.9+vio; % DAV of next period, may not be an integer
% Linear interpolation on DAV
A = [ceil(DAVgrid_new*2)+1,20*ones(N,1)]; % This matrix is prepared for DAVu
DAVu = min(A,[],2);
DAVl = floor(DAVgrid_new*2)+1;
%% Loop 1
for i=1:N
Vinv_u(i) = OldV(omega1(i),DAVu(i),status(i)+1,2,lag_inv(i)+1); % +1 because matlab counts from 1 instead of
Vinv_l(i) = OldV(omega1(i),DAVl(i),status(i)+1,2,lag_inv(i)+1);
Vnotinv_u(i) = OldV(omega1(i),DAVu(i),status(i)+1,1,lag_inv(i)+1);
Vnotinv_l(i) = OldV(omega1(i),DAVl(i),status(i)+1,1,lag_inv(i)+1);
end

Vinv = Vinv_l + (DAVgrid_new*2+1-DAVl).*(Vinv_u-Vinv_l);
Vnotinv = Vnotinv_l + (DAVgrid_new*2+1-DAVl).*(Vnotinv_u-Vnotinv_l);
NewVtil = log(exp(beta*Vinv-Coeff(1))+exp(beta*Vnotinv))+gamma;
Invp = exp(beta*Vinv-Coeff(1))./(exp(beta*Vinv-Coeff(1))+exp(beta*Vnotinv));

for i = 1:N
% Name the variables/vectors
omega1 = Omegat(i,1); % Fixed state
DAV = Omegat(i,2); % From 0 to 19
status = Omegat(i,3); % Violation status
lag_inv = Omegat(i,4); % Lag 1 quarter investment
vio = Omegat(i,5); % Violation
    if status ==0
            % If the plant is in compliance, then at the beginning of next
            % period, DAV = 0, status is compliance, two lags of investment
            % are also 0.
            NewVtil(i) = beta*OldV(omega1,1,1,1,1) + gamma;
            Vtilde(omega1,DAV*2+1,status+1,lag_inv+1,vio+1) = NewVtil(i);
    else
        % Update Vtilde
        Vtilde(omega1,DAV*2+1,status+1,lag_inv+1,vio+1) = NewVtil(i);
        % Investment probability given the plant in on Omega tilde.
        Investprob(omega1,DAV*2+1,status+1,lag_inv+1,vio+1) = Invp(i);
    end
end

%% Vtilde in Loop 2
 for i = 1:N
 % Vtilde for each potential violator status and violation
 omega1 = Omega(i,1);
        DAV = Omega(i,2);
        lag_inv = Omega(i,4);
```

```
            Vr(i,1,1) = Vtilde(omega1,1,1,1,1);
            Vr(i,2,1) = Vtilde(omega1,1,1,1,1);
            Vr(i,1,2) = Vtilde(omega1,DAV*2+1,2,lag_inv+1,1);
            Vr(i,2,2) = Vtilde(omega1,DAV*2+1,2,lag_inv+1,2);
            Vr(i,1,3) = Vtilde(omega1,DAV*2+1,3,lag_inv+1,1);
            Vr(i,2,3) = Vtilde(omega1,DAV*2+1,3,lag_inv+1,2);
 end

 % 80 regulation actions in terms of inspection, violation and fines
 for j = 1:80

            % Let vio be an indicator whether there is a violation in
            % regulation action j

               % Expected Vtilde conditional on Omega and inspection,
            % violation and fines
            if j<=20 || (j>=41 && j<=60) % vio = 0
            EVr(:,j) = dot(Vr(:,1,:),Tran(:,j,:),3);
            else % vio = 1
            EVr(:,j) = dot(Vr(:,2,:),Tran(:,j,:),3);
            end
 end



%% Loop through states Omega. For each Omega, find V(Omega) using equation
    for i = 1:N
        % A1
        omega1 = Omega(i,1);
        DAV = Omega(i,2);
        status = Omega(i,3);
        lag_inv = Omega(i,4);
        lag2_inv = Omega(i,5);

        % Then we can calculate V(Omega) as the sum of expected static
        % utility and expected Vtilde conditional on Omega being the
        % initial state
        NewV(omega1,DAV*2+1,status+1,lag_inv+1,lag2_inv+1) =...
            dot(EVr(i,:),P_regu(i,:))+EEU(i);
    end

norm = max(abs(NewV - OldV),[],'all');
OldV = NewV; % Update value function

%disp([norm,iter])
end
```

```
toc
end
```

LogLike:

```
    function [loglike,ll] = LogLike(Investprob,somega)
% Number of observations
N = size(somega,1);
loglike = 0;
ll = zeros(N,1);
for i = 1:N
    omega1 = somega.omega1(i);
    DAV = somega.DAV(i);
    DAVh = min(ceil(DAV*2)+1,20);
    DAVl = floor(DAV*2)+1;
    status = somega.ordered_violator(i)+1;
    laginv = somega.lag_investment(i)+1;
    vio = somega.violation(i)+1;
    X = somega.investment(i);
    Likelihoodh = X*Investprob(omega1,DAVh,status,laginv,vio) +...
        (1-X)*(1-Investprob(omega1,DAVh,status,laginv,vio));
    Likelihoodl = X*Investprob(omega1,DAVl,status,laginv,vio) +...
        (1-X)*(1-Investprob(omega1,DAVl,status,laginv,vio));
    Likelihood = Likelihoodl + (DAV*2+1-DAVl)*(Likelihoodh-Likelihoodl);
    ll(i) = log(Likelihood);
    loglike = loglike + log(Likelihood);
end
end
```

loglikefun:

```
    function loglike = loglikefun(Theta,Omega,Omegat,Bellman,somega)
%LOGLIKEFUN Summary of this function goes here
%   Detailed explanation goes here
[~,~,Investprob]=Bellmanfun(Theta,Omega,Omegat,Bellman);
[loglike,~] = LogLike(Investprob,somega);
end
```

VarML:

```
    function VarML = varml(x,fval,Omega,Omegat,Bellman,somega)
% Initialize the derivative of loglikelihood with respect to theta
N = size(somega,1);
dlogl = zeros(N,5);
I = eye(5)*1e-6;
```

```
for j = 1:5
    % plus 1e-6 to the jth element in theta
    newtheta = x + I(:,j);
    [~,~,Investprob]=Bellmanfun(newtheta,Omega,Omegat,Bellman);
    [~,ll] = LogLike(Investprob,somega);
    dlogl(:,j) = ll+fval;% fval is - loglikelihood when theta is theta_ML

end
VarML = (dlogl.'*dlogl)^(-1);
end
```