

Homework 5

Kensuke Suzuki

November 28, 2018

Problem 1: Gaussian Quadrature

In this problem, I rule out u_i from the model. Define the function `llk_wou(Y,X,Z,par,node,method)` which returns (negative of) the log likelihood, given data (X, Y, and Z), parameter vector (par), number of node (node), and specified integration method (method).

In the first problem, I use the Gaussian quadrature method; `method=1` with 20 nodes. Using `qwnorm()` included in the CEtools, I draw 20 nodes for β_i from the normal distribution with mean β_0 and variance σ_β^2 . This also generates the weighting vector `w` which I use later. I pick each draw of β_i , compute the likelihood for each i , $L_i(\gamma|\beta_i, u_i)$, and stack it up for all draws. Numerical integration is completed by calculating the weighted average of the likelihood using the weights obtained above. Finally take log and sum over all i . **Log likelihood is $-1.2571e + 03$.**

Problem 2: Monte Carlo

In the second problem, I use the Monte Carlo method; `method=2` with 100 nodes. I draw 100 nodes using `haltonNormshuddle()` provided in the lecture. Analogous to the first problem, for each draw, I compute the likelihood $L_i(\gamma|\beta_i, u_i)$, stack it up for all draws, and compute the simple average. Finally take log and sum over all i . **Log likelihood is $-1.2571e + 03$.**

Problem 3: MLE without u_i using `fmincon`

We use `fmincon` to estimate the parameters. Let parameter vector $\theta = [\gamma_0 \ \beta_0 \ \sigma_\beta^2]'$. We need to impose the parameter restrictions such that $\sigma_\beta^2 \geq 0$. We define $\mathbf{A} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}$ and $b = 0$. When minimizing the negative of the log likelihood over the parameter vector θ , I have a constraint $\mathbf{A}\theta \leq b$. Results are presented below:

Gaussian Quadrature

$$\text{Initial guess: } \begin{bmatrix} \gamma_0 \\ \beta_0 \\ \sigma_\beta^2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \text{ estimates: } \begin{bmatrix} \hat{\gamma} \\ \hat{\beta} \\ \hat{\sigma}_\beta^2 \end{bmatrix} = \begin{bmatrix} -0.5056 \\ 2.4832 \\ 1.4054 \end{bmatrix}, \text{ loglikelihood: } -536.2378$$

Monte Carlo

$$\text{Initial guess: } \begin{bmatrix} \gamma_0 \\ \beta_0 \\ \sigma_\beta^2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \text{ estimates: } \begin{bmatrix} \hat{\gamma} \\ \hat{\beta} \\ \hat{\sigma}_\beta^2 \end{bmatrix} = \begin{bmatrix} -0.5056 \\ 2.5578 \\ 1.1816 \end{bmatrix}, \text{ loglikelihood: } -536.5876$$

Matlab function llk_wou()

```

1 function [llf,methodname] = llk_wou(Y,X,Z,par,node,method)
2     % compute negative of llf
3
4     gamma = par(1);
5     betanot = par(2);
6     sigmab = par(3);
7     %unot = par(4);
8     unot = 0;
9     %sigmaub = par(5);
10    sigmaub = 0;
11    %sigmau = par(6);
12    sigmau = 0;
13
14    mu = [betanot unot];
15    Sigma = [sigmab sigmaub; sigmaub sigmau];
16
17    ui = 0;
18
19
20    if method == 1
21        methodname = 'Gaussian Quadrature';
22        % if method is Gaussian Quadrature
23        [rcoef,w] = qnwnorm(node, betanot, sigmab);
24
25
26    for i = 1:length(rcoef)
27        betai = rcoef(i,1);
28        % pick ith draw of beta
29        epsi = (betai * X + gamma * Z + ui) ;
30        logitval = ( 1 + exp(-1 * epsi) ).^(-1);
31        % compute the logistic CDF
32        lkt = logitval.^Y .* (ones(20,100)-logitval).^((ones(20,100)-Y));
33        % compute the contribution of each year

```

```

34     lkii(i,:) = prod(lkt);
35         % product over years
36 end
37 lki = w' * lkii;
38     % numerical integration
39
40 llki = log(lki);
41
42 llf = -1 * sum(llki,2);
43
44 elseif method == 2
45     methodname = 'Monte Carlo';
46         % if method is MC
47
48     norm = haltonNormShuffle(node, 1, 3);
49     rcoef = repmat(betanot,node,1) + sigmab * norm';
50
51 for i = 1:length(rcoef)
52     betai = rcoef(i,1);
53         % pick ith draw of beta
54     epsi = (betai * X + gamma * Z + ui ) ;
55     logitval = ( 1 + exp(-1 * epsi) ).^(-1);
56         % compute the logistic CDF
57     lkt = logitval.^Y .* (ones(20,100)-logitval).^((ones(20,100)-Y));
58         % compute the contribution of each year
59     lkii(i,:) = prod(lkt);
60         % product over years
61 end
62 lki = sum(1/node * lkii);
63     % numerical integration
64
65 llki = log(lki);
66
67 llf = -1 * sum(llki,2);
68
69 end
70
71
72 end

```

Problem 4: MLE of full model using fmincon

In this problem, I estimate the full model. Define the function `llk_wu(Y,X,Z,par,node,method)` which returns (negative of) the log likelihood of the full model, given data (X, Y, and Z), parameter vector (par), number of node (node), and specified integration method (method). Since I only invoke Monte Carlo method, `method=2`.

In this function, I draw 100 nodes using `haltonNormshuddle()`. For this time, I draw β_i and u_i from the bivariate normal distribution with mean $\mu = [\beta_0, u_0]'$ and variance-covariance matrix

$\Sigma = \begin{bmatrix} \sigma_\beta & \sigma_{u\beta} \\ \sigma_{u\beta} & \sigma_u \end{bmatrix}$. I use `chol()` to make Cholesky decomposition of Σ to simulate from the joint distribution—bivariate normal. Implementation of numerical integration is same as in Problem 2. For optimization, I use `fmincon`. In addition to the nonnegative restrictions on variances, σ_β^2 and σ_u^2 , I need to restrict the variance-covariance matrix Σ to be positive definite. Therefore, rather than optimizing over $\sigma_{u\beta}$, I optimize over the correlation coefficient ρ with restriction $-1 \leq \rho \leq 1$ and recover $\sigma_{ub} = \rho \sqrt{\sigma_\beta^2} \sqrt{\sigma_u^2}$. As I have learned in the lecture, I may constrain ρ by imposing upper and lower bound. But since there is no special reason here, I simply write the constraint by linear equation.

Parameter vector over which I optimize is $\theta = [\gamma_0 \ \beta_0 \ u_0 \ \sigma_\beta^2 \ \rho \ \sigma_u^2]$. Constraints can be expressed by $A\theta \leq \mathbf{b}$ where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

I will present the initial values and estimated values for $\sigma_{u\beta}$ which is recovered from the ones for ρ . Our initial guess on ρ is 0.9 and estimated values are $\hat{\rho} = 0.4536$.

$$\text{Initial guess: } \begin{bmatrix} \gamma_0 \\ \beta_0 \\ u_0 \\ \sigma_\beta^2 \\ \sigma_{ub} \\ \sigma_u^2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0.9783 \\ 1 \end{bmatrix}, \quad \text{estimates: } \begin{bmatrix} \hat{\gamma} \\ \hat{\beta} \\ \hat{u} \\ \hat{\sigma}_\beta^2 \\ \hat{\sigma}_{ub} \\ \hat{\sigma}_u^2 \end{bmatrix} = \begin{bmatrix} -0.6815 \\ 3.1877 \\ 1.4710 \\ 1.9226 \\ 0.8068 \\ 1.6458 \end{bmatrix}, \quad \text{loglikelihood: } -464.0001$$

Matlab function `llk_wu()`

```
1 function [llf,methodname] = llk_wu(Y,X,Z,par,node,method)
2     % compute negative of llf
3
4     gamma = par(1);
5     betanot = par(2);
6     sigmab = par(3);
7     unot = par(4);
8     %sigmaub = par(5);
9     rho = par(5);
10    sigmau = par(6);
11
12    sigmaub = sigmab^(1/2) * sigmau^(1/2) * rho;
13
14    mu = [betanot unot];
15    Sigma = [sigmab sigmaub; sigmaub sigmau];
16    U = chol(Sigma);
17
```

```
18 if method == 2
19     methodname = 'Monte Carlo';
20     % if method is MC
21
22     norm = haltonNormShuffle(node, 2, 2);
23     rcoef = repmat(mu,node, 1) + (U' * norm)';
24
25 for i = 1:length(rcoef)
26     betai = rcoef(i,1);
27     ui = rcoef(i,2);
28     % pick ith draw of beta
29     epsi = (betai * X + gamma * Z + ui) ;
30     logitval = ( 1 + exp(-1 * epsi) ).^(-1);
31     % compute the logistic CDF
32     lkt = logitval.^Y .* (ones(20,100)-logitval).^(ones(20,100)-Y);
33     % compute the contribution of each year
34     lkii(i,:) = prod(lkt);
35     % product over years
36 end
37 lki = sum(1/node * lkii);
38 % numerical integration
39
40 llki = log(lki);
41
42 llf = -1 * sum(llki,2);
43
44 end
45
46
47 end
```

Matlab Main Code

```
1 % Empirical Method HW5 %
2 % Ken Suzuki (Penn State)
3 % kxs974@psu.edu
4
5 clear all
6 delete HW5log.txt
7 diary('HW5log.txt')
8 diary on
9
10 % Load Data
11 load('hw5.mat')
12
13 X = data.X;
14 Y = data.Y;
15 Z = data.Z;
```

```
16
17 addpath( '.. / CTools / ');
18
19 %% Problem 1
20 % parameter value
21 betanot = 0.1;
22 sigmab = 1;
23 gamma = 0;
24
25 % method
26 method = 1;
27
28 % number of nodes
29 node = 20;
30
31 % set parameter vector
32 par = [gamma betanot sigmab];
33
34 [llf ,methodname] = llk_wou(Y,X,Z,par,node,method);
35
36 llf = -1 * llf;
37 % display result
38 disp('Problem 1')
39 disp(methodname)
40 disp('Loglikelihood is ')
41 disp(llf)
42
43 %% Problem 2
44 % parameter value
45 betanot = 0.1;
46 sigmab = 1;
47 gamma = 0;
48
49 % method
50 method = 2;
51
52 % number of nodes
53 node = 100;
54
55 % set parameter vector
56 par = [gamma betanot sigmab];
57
58 [llf ,methodname] = llk_wou(Y,X,Z,par,node,method);
59 llf = -1 * llf; % take negative
60
61 % display result
62 disp('Problem 2')
63 disp(methodname)
```

```
64 disp('Loglikelihood is ')
65 disp(llf)
66
67 %% Problem 3
68
69 clear par
70
71 % number of node
72 node = 20;
73
74 % method: GC
75 method = 1;
76
77 % define function to be minimized (function of par)
78 llkwou_min = @(par) llkwou(Y,X,Z,par,node,method);
79
80 % fmincon
81
82 % for constraint
83 A = [0, 0, -1];
84 b = 0
85
86 %[paraGQ, lfGQ] = fminsearch(llkwou_min, [1 1 1] );
87 [paraGQ, lfGQ] = fmincon(llkwou_min, [1; 1; 1], A, b );
88 lfGQ = -1 * lfGQ;
89
90 % number of node
91 node = 100;
92
93 % method: GC
94 method = 2;
95
96 % define function to be minimized (function of par)
97 llkwou_min = @(par) llkwou(Y,X,Z,par,node,method);
98
99 % fminsearch
100 [paraMC, lfMC] = fmincon(llkwou_min, [1; 1; 1], A, b );
101 lfMC = -1 * lfMC;
102
103
104 % display result
105 disp('Problem 3-1 (Gaussian Quadrature)')
106 disp('Initial guesses are')
107 disp('    gamma    beta    sigmab')
108 disp([1 1 1])
109 disp('    gamma    beta    sigmab')
110 disp(paraGQ')
111 disp('Maximized log-likelihood is:')
```

```
112 disp(lfGQ)
113
114
115 % display result
116 disp('Problem 3-2 (Monte Carlo)')
117 disp('Initial guesses are')
118 disp('    gamma    beta    sigmab')
119 disp([1 1 1])
120 disp('    gamma    beta    sigmab')
121 disp(paraMC')
122 disp('Maximized log-likelihood is:')
123 disp(lfMC)
124
125
126 %% Problem 4
127
128 clear A
129 clear b
130
131 % method: MC
132 method = 2;
133
134 % number of node
135 node = 100;
136
137 % initial values for parameter vector
138 gamma = -0.5056;
139 betanot = 2.5579;
140 sigmab = 1.1816;
141 unot = 1;
142 %sigmaub = 0.9;
143 rho = 0.9;
144 sigmau = 1;
145 %intpar = [gamma betanot sigmab unot sigmaub sigmau];
146 intpar = [gamma betanot sigmab unot rho sigmau];
147
148 %define function to be minimized
149 llkwu_min = @(par) llk_wu(Y,X,Z,par,node,method);
150
151 % for constraint
152 A = [0 0 -1 0 0 0 ; ...
153      0 0 0 0 0 -1; ...
154      0 0 0 0 -1 0; ...
155      0 0 0 0 1 0];
156 b = [0; 0; 1; 1];
157
158 % fmincon
159 [paraMC, lfMC] = fmincon(llkwu_min, intpar', A, b );
```



```
160
161 sigmaub = paraMC(3)^(1/2) * paraMC(6)^(1/2) * paraMC(5);
162 paraMC_cov = paraMC;
163 paraMC_cov(5) = sigmaub;
164
165 sigmaubint = intpar(3)^(1/2) * intpar(6)^(1/2) *intpar(5);
166 intpar_cov = intpar;
167 intpar_cov(5) = sigmaubint ;
168
169 lfMC = -1 * lfMC;
170
171 % display result
172 disp('Problem 4 (Monte Carlo)')
173 disp('Initial guesses are')
174 disp('    gamma    betanot    sigmab    unot    rho    sigmau')
175 disp(intpar_cov)
176 disp('Estimated parameters')
177 disp('    gamma    betanot    sigmab    unot    sigmaub    sigmau')
178 disp(paraMC_cov)
179 disp('Maximized log-likelihood is:')
180 disp(lfMC)
181
182 diary off
```