Problem Set 2 Solutions

Tyler Ransom

1 Main Problem Set Code

```
clear all;
clc;
load nlsy97m
options = optimset('Disp','iter-detailed','MaxFunEvals',1e12,'MaxIter',1e6);
options2 = ...
    optimset('Disp','iter-detailed','MaxFunEvals',1e12,'MaxIter',1e6,'GradObj','on','Deriva
options3 = ...
    optimset('Disp','iter-detailed','MaxFunEvals',1e12,'MaxIter',1e6,'GradObj','on','Hessia
rand('seed',1234); randn('seed',1234);
%% Reshape the data into long format
N = length(ID);
T = 5;
activityt = activity';
log_waget = log_wage';
hgct = hgc';
expert = exper';
Diplomat = Diploma';
AAt = AA';
BAt = BA';
malet = repmat(male,1,T)';
AFQTt = repmat(AFQT,1,T)';
Mhgct = repmat(Mhgc,1,T)';
IDt = repmat(ID,1,T)';
IDl = IDt(:);
IDprime  = [1:numel(ID)]';
IDprimet = repmat(IDprime,1,T)';
IDprimel = IDprimet(:);
y = activityt(:);
log_wagel = log_waget(:);
X = [ones(N*T,1) malet(:) AFQTt(:) Mhgct(:) hgct(:) expert(:) Diplomat(:) ...
    AAt(:) BAt(:)];
K = size(X,2);
%% Problem 2(a)
tabulate(activityt(:));
%% Problem 2(b) -- estimate mlogit using fminunc
bstart = rand(5*size(X,2),1);
tic;
[bng,lng,~,~,~,hng] = fminunc('mlogit_gradient',bstart(:),options,X,y);
SEbng = sqrt(diag(inv(hng)));
```

```
disp('Time spent for mlogit MLE (no gradient)')
toc;
resultng = [bng SEbng]
%% Problem 2(c) -- estimate mlogit using fminunc (and provide gradient)
tic;
[bg,lg,~,~,~,hg] = fminunc('mlogit_gradient',bstart(:),options2,X,y);
hg = full(hg);
SEbg = sqrt(diag(inv(hg)));
disp('Time spent for mlogit MLE (user-supplied gradient)')
toc;
resultg = [bg SEbg]
%% Problem 3(a) -- log wage normal MLE
subset = ~isnan(log_waget(:));
X = [ones(N*T,1) malet(:) AFQTt(:) Mhgct(:) hgct(:) expert(:) Diplomat(:) ...
    AAt(:) BAt(:) y==2 y==3];
X1 = X(subset,:);
y1 = log_wagel(subset);
bols = (X1'*X1)\(X1'*y1);
[b,l,~,~,~,h]=fminunc('normalMLEconstraint',-.45*rand(length(bols)+1,1),options,X1,y1);
b(2)   = .15*tanh(b(2))+.05;
b(10)  = -(b(10)).^2;
b(end) = exp(b(end));

A = ones(length(bols)+1,1);
A(2)   = .15*(sech(b(2)).^2);
A(10)  = -2*(b(10));
A(end) = exp(b(end));
A = diag(A);

SE = sqrt(diag(A'/(h)*A));

resultMLE = [b SE]
%% Problem 4(a) -- log wage normal MLE with gradient
b0 = rand(length(bols)+1,1);
[bmleng,lmleng,~,~,~,hmleng]=fminunc('normalMLEgradient',b0,options,X1,y1);
SEmleng = sqrt(diag(inv(hmleng)));
resultMLEng = [bmleng SEmleng]
[bmleg,lmleg,~,~,~,hmleg]   =fminunc('normalMLEgradient',b0,options2,X1,y1);
hmleg = full(hmleg);
SEmleg = sqrt(diag(inv(hmleg)));
resultMLEg = [bmleg SEmleg]
%% Problem 4(b) -- log wage normal MLE with hessian
[bmlenh,lmlenh,~,~,~,hmlenh]=fminunc('normalMLEhessian',b0,options,X1,y1);
SEmlenh = sqrt(diag(inv(hmlenh)));
resultMLEnh = [bmlenh SEmlenh]
[bmleh,lmleh,~,~,~,hmleh]   =fminunc('normalMLEhessian',b0,options3,X1,y1);
SEmleh = sqrt(diag(inv(hmleh)));
resultMLEh = [bmleh SEmleh]
save PS2results resultg lg resultng lng resultMLE l resultMLEng lmleng ...
    resultMLEg lmleg resultMLEnh lmlenh resultMLEh lmleh
```

2 MLE Function Code

```
function [like,grad] = mlogit_gradient(b,X,y)
b1 = b(1+0*size(X,2):1*size(X,2));
b2 = b(1+1*size(X,2):2*size(X,2));
b3 = b(1+2*size(X,2):3*size(X,2));
b4 = b(1+3*size(X,2):4*size(X,2));
b5 = b(1+4*size(X,2):5*size(X,2));
dem = 1+exp(X*b1)+exp(X*b2)+exp(X*b3)+exp(X*b4)+exp(X*b5);
like = -sum((y==1).*(X*b1) + (y==2).*(X*b2) + (y==3).*(X*b3) + ...
    (y==4).*(X*b4) + (y==5).*(X*b5) + (y==6) - log(dem));
P1 = exp(X*b1)./dem;
P2 = exp(X*b2)./dem;
P3 = exp(X*b3)./dem;
P4 = exp(X*b4)./dem;
P5 = exp(X*b5)./dem;
grad = zeros(size(b));
grad(1+0*size(X,2):1*size(X,2),1) = -X'*((y==1)-P1);
grad(1+1*size(X,2):2*size(X,2),1) = -X'*((y==2)-P2);
grad(1+2*size(X,2):3*size(X,2),1) = -X'*((y==3)-P3);
grad(1+3*size(X,2):4*size(X,2),1) = -X'*((y==4)-P4);
grad(1+4*size(X,2):5*size(X,2),1) = -X'*((y==5)-P5);
```

```
function ll = normalMLEconstraint(theta,x,y)
% b = theta(1:end-1);
theta(2)   = .15*tanh(theta(2))+.05;
theta(10)  = -(theta(10)).^2;
theta(end) = exp(theta(end));

ll = ...
   -sum(-.5*log(2*pi*(theta(end)^2))-.5*((y-x*theta(1:end-1))/theta(end)).^2);
end
```

```
function [like,grad]=normalMLEgradient(b,X,Y)
beta      = b(1:end-1);
wagesigma = b(end);
n         = length(Y);

like = -sum(-.5*log(2*pi)-.5*log(wagesigma^2)-.5*((Y-X*beta)./wagesigma).^2);

grad = zeros(size(b));
grad(1:end-1) = -X'*(Y-X*beta)./wagesigma^2;
grad(end)     = n./wagesigma - ((Y-X*beta)'*(Y-X*beta))./wagesigma^3;
end
```

```
function [like,grad,hess]=normalMLEhessian(b,X,Y)
beta      = b(1:end-1);
wagesigma = b(end);
```

```
n          = length(Y);
like = -sum(-.5*log(2*pi)-.5*log(wagesigma^2)-.5*((Y-X*beta)./wagesigma).^2);

grad = zeros(size(b));
grad(1:end-1) = -X'*(Y-X*beta)./wagesigma^2;
grad(end)     = n./wagesigma - ((Y-X*beta)'*(Y-X*beta))./wagesigma^3;

hess = zeros(length(b));
hess(1:length(b)-1,1:length(b)-1) = (X'*X)./wagesigma.^2;
hess(length(b),1:length(b)-1) = (2*(Y-X*beta)'*X)./wagesigma.^3;
hess(1:length(b)-1,length(b)) = (2*X'*(Y-X*beta))./wagesigma.^3;
hess(length(b),length(b)) = (-n)./wagesigma.^2 + ...
    (3./wagesigma.^4)*(Y-X*beta)'*(Y-X*beta);
end
```