Problem Set 2 Solutions

Tyler Ransom

June 7, 2012

1 Main Problem Set Code

```
clear all;
clc;
load nlsw88
options = optimset('Disp','iter-detailed','MaxFunEvals',1e12,'MaxIter',1e6);
rand('seed',1234); randn('seed',1234);
%% Problem 1(a)
%%% Set up the data:
y = log(wage);
X = [ones(size(wage)) age race==2 race==3 collgrad];
%%% create a vector that is 1 if all obs are there; 0 otherwise:
subset = ¬...
    isnan(wage)&¬isnan(age)&¬isnan(race)&¬isnan(married)&¬isnan(grade)&¬isnan(collgrad);
y = log(wage(subset)); %drop missing observations from y
X = X(subset,:);       %drop missing observations from X
nb = size(X,2);        %initialize the number of regressors for later use


% (i)   Estimate \hat{\beta}  and s^{2}  (variance of \varepsilon_{i} ) ...
    using fminsearch with default convergence tolerances
[bOLSsearch,SSEsearch]=fminsearch('OLS1',rand(nb,1),options,X,y);
s2OLSsearch = SSEsearch/(length(y)-size(X,2));


% (ii)  Estimate \hat{\beta}  and s^{2}  (variance of \varepsilon_{i} ) ...
    using fminunc with default convergence tolerances
[bOLSunc,SSEunc]=fminunc('OLS1',rand(nb,1),options,X,y);
s2OLSunc = SSEunc/(length(y)-size(X,2));


% (iii) Estimate \hat{\beta}  and s^{2}  (variance of \varepsilon_{i} ) ...
    using fmincon with default convergence tolerances and with \beta_{3}<0 ...
     as the only restriction
%%% Initialize fmincon constraints to be empty (i.e. no constraint ...
    inputs), except for an upper bound on beta3
A = [];
b = [];
```

```
Aeq = [];
beq = [];
lb = [];
ub = 10*ones(size(X,2),1);
ub(3) = 0; %(set the upper bound at 0 for beta3, but no upper bound for ...
    the rest)
nonlcon = [];
[bOLScon,SSEcon]=fmincon('OLS1',rand(nb,1),A,b,Aeq,beq,lb,ub,nonlcon,options,X,y);
s2OLScon = SSEcon/(length(y)-size(X,2));


% (iv)  How do your answers differ when using each of the optimizers?
ansA = (X'*X)\X'*y;
SSEansA = (y-X*ansA)'*(y-X*ansA);
ansA = [ansA; (y-X*ansA)'*(y-X*ansA)/(length(y)-size(X,2))];
compareA = [[bOLSsearch;s2OLSsearch;SSEsearch] [bOLSunc;s2OLSunc;SSEunc] ...
    [bOLScon;s2OLScon;SSEcon] [ansA;SSEansA]]
%% Problem 1(b)


% (i)   Estimate \hat{\beta}  and \hat{\sigma}^{2}  (variance of ...
    \varepsilon_{i} ) using fminsearch with default convergence tolerances
[bMLEsearch,likeSearch]=fminsearch('normalMLE',rand(nb+1,1),options,X,y);
bMLEsearch(end) = bMLEsearch(end)^2; %recover sigma2, not sigma


% (ii)  Estimate \hat{\beta}  and \hat{\sigma}^{2}  using fminunc with ...
    default convergence tolerances
[bMLEunc,likeUnc]=fminunc('normalMLE',rand(nb+1,1),options,X,y);
bMLEunc(end) = bMLEunc(end)^2;


% (iii) Estimate \hat{\beta}  and \hat{\sigma}^{2}  using fmincon with ...
    default convergence tolerances and with \beta_{3}<0  as the only ...
    restriction
lb = -10*ones(size(X,2)+1,1);
lb(end)=0;
ub = 10*ones(size(X,2)+1,1);
[bMLEcon,likeCon]=fmincon('normalMLE',.5*rand(nb+1,1)-.25,A,b,Aeq,beq,lb,ub,nonlcon,option
bMLEcon(end) = bMLEcon(end)^2;


% (iv)  How sensitive is \hat{\beta}  to the normal distribution ...
    assumption? How close are s^{2}  and \hat{\sigma}^{2}
compareA
compareB = [[bMLEsearch;-likeSearch] [bMLEunc;-likeUnc] [bMLEcon;-likeCon] ...
    [ansA;SSEansA]]
%% Problem 1(c)
%%% Set up the data:
y = log(wage);
X = [ones(size(wage)) age race==2 race==3 collgrad grade married south...
     c_city union ttl_exp tenure age.^2 hours never_married];
%%% create a vector that is 1 if all obs are there; 0 otherwise:
subset1 = ¬isnan(wage)&¬isnan(age)&¬isnan(race)&¬isnan(married)...
```

```
         &¬isnan(grade)&¬isnan(collgrad)&¬isnan(south)&¬isnan(c_city)...
         &¬isnan(union)&¬isnan(ttl_exp)&¬isnan(tenure)&¬isnan(hours)...
         &¬isnan(never_married);
y = log(wage(subset1)); %drop missing observations from y
X = X(subset1,:);       %drop missing observations from X
nb = size(X,2);         %initialize the number of regressors for later use
%%% Initialize the baseline closed-form OLS formulas (for later comparison)
ansC = (X'*X)\X'*y;
sseC = (y-X*ansC)'*(y-X*ansC);
s2C = (y-X*ansC)'*(y-X*ansC)/(length(y)-size(X,2));
%%% Initialize the width of the OLS+noise for starting values:
alpha = .75;


% (i)   Estimate \hat{\beta}  and s^{2}  using fminsearch with default ...
    convergence tolerances, assuming \varepsilon_{i}  is mean-zero
% I'm going to write a function called OLS1 that minimizes the sum of the ...
    squared errors for any vector Y and matrix X
[bOLSsearch,SSEsearch,eOLSsearch]=fminsearch('OLS1',ansC+(2*alpha*ansC.*rand(size(ansC))-al
s2OLSsearch = SSEsearch/(length(y)-size(X,2));


% (ii)  Estimate \hat{\beta}  and s^{2}  using fminunc with default ...
    convergence tolerances, assuming \varepsilon_{i}  is mean-zero
[bOLSunc,SSEunc,eOLSunc]=fminunc('OLS1',ansC+(2*alpha*ansC.*rand(size(ansC))-alpha*ansC),op
s2OLSunc = SSEunc/(length(y)-size(X,2));


% (iii) Estimate \hat{\beta}  and \hat{\sigma}^{2}  using fminsearch with ...
    default convergence tolerances, assuming ...
    \varepsilon_{i}\overset{iid}{\sim}N\left(0,\sigma\right)
[bMLEsearch,likeSearch,eMLEsearch]=fminsearch('normalMLE',[ansC+(2*alpha*ansC.*rand(size(an
bMLEsearch(end) = bMLEsearch(end)^2; %recover sigma2, not sigma


% (iv)  Estimate \hat{\beta}  and \hat{\sigma}^{2}  using fminunc with ...
    default convergence tolerances, assuming ...
    \varepsilon_{i}\overset{iid}{\sim}N\left(0,\sigma\right)
[bMLEunc,likeUnc,eMLEunc]=fminunc('normalMLE',[ansC+(2*alpha*ansC.*rand(size(ansC))-alpha*a
bMLEunc(end) = bMLEunc(end)^2;
% [bMLEunc,likeUnc,eMLEunc]=fminunc('normalMLE',bMLEunc,options,X,y);
% bMLEunc(end) = bMLEunc(end)^2;


% (v)   How does fminsearch compare to fminunc when the dimension of the ...
    parameter vector increases?
exitFlags = [eOLSsearch eOLSunc eMLEsearch eMLEunc]
compareC = [[bOLSsearch;s2OLSsearch;SSEsearch] [bOLSunc;s2OLSunc;SSEunc] ...
    [bMLEsearch;-likeSearch] [bMLEunc;-likeUnc] [ansC; s2C; sseC]]
compareA
compareB


% (vi)  How do the estimators in (i) through (iv) perform when the ...
```

```
    starting values are 0.01 for all parameters?
[bOLSsearch,SSEsearch,eOLSsearch]=fminsearch('OLS1',.01*ones(nb,1),options,X,y);
s2OLSsearch = SSEsearch/(length(y)-size(X,2));

[bOLSunc,SSEunc,eOLSunc]=fminunc('OLS1',.01*ones(nb,1),options,X,y);
s2OLSunc = SSEunc/(length(y)-size(X,2));

[bMLEsearch,likeSearch,eMLEsearch]=fminsearch('normalMLE',.01*ones(nb+1,1),options,X,y);
bMLEsearch(end) = bMLEsearch(end)^2; %recover sigma2, not sigma

[bMLEunc,likeUnc,eMLEunc]=fminunc('normalMLE',.01*ones(nb+1,1),options,X,y);
bMLEunc(end) = bMLEunc(end)^2;
exitFlags2 = [eOLSsearch eOLSunc eMLEsearch eMLEunc]
compareC2 = [[bOLSsearch;s2OLSsearch;SSEsearch] [bOLSunc;s2OLSunc;SSEunc] ...
    [bMLEsearch;-likeSearch] [bMLEunc;-likeUnc] [ansC; s2C; sseC]]
compareC
%% Problem 2(a)
clear all;
load nhanes2d
options = optimset('Disp','iter-detailed','MaxFunEvals',1e12,'MaxIter',1e6);
options2 = ...
    optimset('Disp','iter-detailed','MaxFunEvals',1e12,'MaxIter',1e6,'TolX',1e-8,'TolFun',1
%%% Set up the data:
y = hct;
X = [ones(size(hct)) age race==2 race==3 heartatk sex==2 highbp region==1 ...
     region==2 region==3 smsa==2 smsa==4 height weight houssiz];
%%% create a vector that is 1 if all obs are there; 0 otherwise:
subset2 = ¬isnan(hct)&¬isnan(age)&¬isnan(race)&¬isnan(heartatk)...
          &¬isnan(sex)&¬isnan(highbp)&¬isnan(region)&¬isnan(smsa)...
          &¬isnan(height)&¬isnan(weight)&¬isnan(houssiz);
y = hct(subset2); %drop missing observations from y
X = X(subset2,:); %drop missing observations from X
nb = size(X,2);   %initialize the number of regressors for later use
%%% Initialize the baseline closed-form OLS formulas (for later comparison)
ans2A = (X'*X)\X'*y;
sse2A = (y-X*ans2A)'*(y-X*ans2A);
s22A = (y-X*ans2A)'*(y-X*ans2A)/(length(y)-size(X,2));
%%% Initialize the width of the OLS+noise for starting values:
alpha = 1.5;


% i. Estimate \hat{\beta}  and \hat{\sigma}^{2}  using fminsearch with ...
    default convergence tolerances, assuming ...
    \varepsilon_{i}\overset{iid}{\sim}N\left(0,\sigma\right) . Use the ...
    same starting values as in part (i) of 1(c), but now with \alpha=1.5 .
[bMLEsearch1,likeSearch1,eMLEsearch1]=fminsearch('normalMLE',[ans2A+(2*alpha*ans2A.*rand(s
bMLEsearch1(end) = bMLEsearch1(end)^2; %recover sigma2, not sigma


% ii. Estimate \hat{\beta}  and \hat{\sigma}^{2}  using fminsearch with ...
    TolX and TolFun each set to 10^{-8}  (instead of the default), ...
    assuming \varepsilon_{i}\overset{iid}{\sim}N\left(0,\sigma\right) . ...
    Use the same starting values as in part (i) of 2(a).
[bMLEsearch2,likeSearch2,eMLEsearch2]=fminsearch('normalMLE',[ans2A+(2*alpha*ans2A.*rand(s
```

```
bMLEsearch2(end) = bMLEsearch2(end)^2;


% iii. Estimate \hat{\beta}  and \hat{\sigma}^{2}  using fminunc with ...
    default convergence tolerances, assuming ...
    \varepsilon_{i}\overset{iid}{\sim}N\left(0,\sigma\right) . Use the ...
    same starting values as in part (i) of 2(a).
[bMLEunc1,likeUnc1,eMLEunc1]=fminunc('normalMLE',[ans2A+(2*alpha*ans2A.*rand(size(ans2A))−a
bMLEunc1(end) = bMLEunc1(end)^2;


% iv. Estimate \hat{\beta}  and \hat{\sigma}^{2}  using fminunc with TolX ...
    and TolFun each set to 10^{−8}  (instead of the default), assuming ...
    \varepsilon_{i}\overset{iid}{\sim}N\left(0,\sigma\right) . Use the ...
    same starting values as in part (i) of 2(a).
[bMLEunc2,likeUnc2,eMLEunc2]=fminunc('normalMLE',[ans2A+(2*alpha*ans2A.*rand(size(ans2A))−a
bMLEunc2(end) = bMLEunc2(end)^2;


% v. How do your answers change when the convergence tolerance changes? ...
    How many more iterations did the optimization require under the ...
    stricter tolerances? How different are your answers depending on the ...
    optimizer?
exitFlags = [eMLEsearch1 eMLEsearch2 eMLEunc1 eMLEunc2]
compareTol = [[bMLEsearch1;−likeSearch1] [bMLEsearch2;−likeSearch2] ...
    [bMLEunc1;−likeUnc1] [bMLEunc2;−likeUnc2] [ans2A; s22A; sse2A]]
```

2 OLS Function Code

```
function SSE = OLS1(beta,X,Y)
SSE = (Y-X*beta)'*(Y-X*beta);
end
```

3 MLE Function Code

```
function [like]=normalMLE(b,X,Y)
beta      = b(1:end-1);
wagesigma = b(end);
like = sum(.5*log(2*pi*(wagesigma^2))+.5*((Y-X*beta)/wagesigma).^2);
end
```