Problem Set 9 Parameter Estimation and Statistics
10.34 Fall 2005, KJ Beers

Ben Wang, Mark Styczynski
December 9, 2005

Problem 1: 8.A.1

We are asked to set up the linear algebraic equations to obtain a least squares parameter estimate. Our linearized model is of the form:

$$\log_{10} Nu = \log_{10} \alpha_0 + \alpha_1 \log_{10} \mathrm{Re} + \alpha_2 \log_{10} \mathrm{Pr} \tag{1}$$

Using our standard notation, our response, $\underline{y}$, is a measured value of *Nu*

$$\underline{y} = \log_{10} Nu = \log[1.9676;0.8986;0.4261;2.5098;1.1521;0.5520] \tag{2}$$

Our input parameters are the values of Re and Pr. Taking the log of these values we use them in our predictor matrix (design matrix) which has the form of:

$$X = \begin{bmatrix} 1 & 0 & -0.3147 \\ 1 & -1 & -0.3147 \\ 1 & -2 & -0.3147 \\ 1 & 0 & 0.4055 \\ 1 & -1 & 0.4055 \\ 1 & -2 & 0.4055 \end{bmatrix} \tag{3}$$

And our parameters that we'd like to fit, according to our linear model are:

$$\underline{\theta} = \begin{bmatrix} \log_{10} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} \tag{4}$$

The system of equations to solve, we get from equation (8.23) from Beers:

$$\left(X^T X\right)\underline{\theta}_{LS} = X^T \underline{y} \tag{5}$$

This looks very similar to our old friend A$\underline{x}$ = $\underline{b}$, where $X^T X = A$ and $X^T \underline{y} = b$.
Now we just use Gaussian elimination to solve this set of equations:

$$\underline{\theta}_{LS} = A \backslash b \tag{6}$$

Working out all the algebra we get the following system of equations:

$$A = X^T X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -1 & -2 & 0 & -1 & -2 \\ -0.3147 & -0.3147 & -0.3147 & 0.4055 & 0.4055 & 0.4055 \end{bmatrix} \begin{bmatrix} 1 & 0 & -0.3147 \\ 1 & -1 & -0.3147 \\ 1 & -2 & -0.3147 \\ 1 & 0 & 0.4055 \\ 1 & 1 & 0.4055 \\ 1 & -2 & 0.4055 \end{bmatrix} =$$

$$\begin{bmatrix} 6 & -6 & 0.1182 \\ -6 & 10 & -0.1182 \\ 0.1182 & -0.1182 & 0.1491 \end{bmatrix}$$

$$b = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -1 & -2 & 0 & -1 & -2 \\ -0.3147 & -0.3147 & -0.3147 & 0.4055 & 0.4055 & 0.4055 \end{bmatrix} \begin{bmatrix} 0.2939 \\ -0.0464 \\ -0.3705 \\ 0.3996 \\ 0.0615 \\ -0.2581 \end{bmatrix} =$$

$$\begin{bmatrix} 0.0801 \\ 1.2420 \\ 0.0526 \end{bmatrix}$$

Plugging these values in, we get to:

$$\begin{bmatrix} 6 & -6 & 0.1182 \\ -6 & 10 & -0.1182 \\ 0.1182 & -0.1182 & 0.1491 \end{bmatrix} \begin{bmatrix} \log_{10} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 0.0801 \\ 1.2420 \\ 0.0526 \end{bmatrix} \tag{7}$$

Solving via Gaussian elimination by hand

$$\begin{bmatrix} 6 & -6 & 0.1182 \\ -6 & 10 & -0.1182 \\ 0.1182 & -0.1182 & 0.1491 \end{bmatrix} \begin{bmatrix} 0.0801 \\ 1.2420 \\ 0.0526 \end{bmatrix}$$

First we determine $\lambda_{21} = \frac{A_{21}}{A_{11}} = -1$. Now we take [row 2] – λ_{21} [row1]

$$\begin{bmatrix} 6 & -6 & 0.1182 \\ 0 & 4 & 0 \\ 0.1182 & -0.1182 & 0.1491 \end{bmatrix} \begin{bmatrix} 0.0801 \\ 1.3221 \\ 0.0526 \end{bmatrix}$$

next we determine $\lambda_{31} = \frac{A_{31}}{A_{11}} = 0.0197$. Now we take [row 3] - λ_{23} [row1]

$$\begin{bmatrix} 6 & -6 & 0.1182 \\ 0 & 4 & 0 \\ 0 & 0 & 0.1468 \end{bmatrix} \begin{bmatrix} 0.0801 \\ 1.3221 \\ 0.0510 \end{bmatrix}$$

We are fortunate that this is all the elimination we need to do. Using backward substitution:

$$\underline{\theta}_{LS} = \begin{bmatrix} \log_{10} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 0.3370 \\ 0.3305 \\ 0.3475 \end{bmatrix} \tag{8}$$

So these are the parameters that minimize the least squared error. Now we are asked to find the 95% confidence intervals. The 95% confidence interval can be calculated using equation (8.132) from Beers:

$$\theta_j = \theta_{M,j} \pm T_{\nu,\alpha/2} s \left\{ \left[X^T X \big|_{\underline{\theta}_M} \right]^{-1}_{jj} \right\}^{1/2} \tag{9}$$

For each of the parameters, all we have to know is the sample variance, the appropriate T-distribution value for a given ν and α, and the diagonals of $(X^T X)^{-1}$.

$(X^T X)^{-1}$ is calculated from $\left(X^T X\right)\left(X^T X\right)^{-1} = I$

$$\begin{bmatrix} 6 & -6 & 0.1182 \\ -6 & 10 & -0.1182 \\ 0.1182 & -0.1182 & 0.1491 \end{bmatrix} \begin{bmatrix} \left(X^T X\right)^{-1}_{11} & \left(X^T X\right)^{-1}_{12} & \left(X^T X\right)^{-1}_{13} \\ \left(X^T X\right)^{-1}_{21} & \left(X^T X\right)^{-1}_{22} & \left(X^T X\right)^{-1}_{23} \\ \left(X^T X\right)^{-1}_{31} & \left(X^T X\right)^{-1}_{33} & \left(X^T X\right)^{-1}_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This is can be rewritten into 3 systems of equations

$$\begin{bmatrix} 6 & -6 & 0.1182 \\ -6 & 10 & -0.1182 \\ 0.1182 & -0.1182 & 0.1491 \end{bmatrix} \begin{bmatrix} \left(X^T X\right)^{-1}_{11} \\ \left(X^T X\right)^{-1}_{21} \\ \left(X^T X\right)^{-1}_{31} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Eliminating with the same λ_{31} λ_{21} we get

$$\begin{bmatrix} 6 & -6 & 0.1182 \\ 0 & 4 & 0 \\ 0 & 0 & 0.1468 \end{bmatrix} \begin{bmatrix} \left(X^T X\right)^{-1}_{11} \\ \left(X^T X\right)^{-1}_{21} \\ \left(X^T X\right)^{-1}_{31} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -0.0197 \end{bmatrix}$$

Backward substitution gets you:

$$\begin{bmatrix} \left(X^T X\right)^{-1}_{11} \\ \left(X^T X\right)^{-1}_{21} \\ \left(X^T X\right)^{-1}_{31} \end{bmatrix} = \begin{bmatrix} 0.4193 \\ 0.25 \\ -0.1342 \end{bmatrix}$$

This can repeated for the other "systems". This is especially easy because we use the same A matrix with the same pivots. Therefore for the following system:

$$\begin{bmatrix} 6 & -6 & 0.1182 \\ -6 & 10 & -0.1182 \\ 0.1182 & -0.1182 & 0.1491 \end{bmatrix} \begin{bmatrix} \left(X^T X\right)^{-1}_{12} \\ \left(X^T X\right)^{-1}_{22} \\ \left(X^T X\right)^{-1}_{32} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

we get:

$$\begin{bmatrix} 6 & -6 & 0.1182 \\ 0 & 4 & 0 \\ 0 & 0 & 0.1468 \end{bmatrix} \begin{bmatrix} \left(X^T X\right)^{-1}_{12} \\ \left(X^T X\right)^{-1}_{22} \\ \left(X^T X\right)^{-1}_{32} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Backward substitution gets you:

$$\begin{bmatrix} \left(X^T X\right)^{-1}_{12} \\ \left(X^T X\right)^{-1}_{22} \\ \left(X^T X\right)^{-1}_{32} \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0.25 \\ 0 \end{bmatrix}$$

The last system yields:

$$\begin{bmatrix} 6 & -6 & 0.1182 \\ -6 & 10 & -0.1182 \\ 0.1182 & -0.1182 & 0.1491 \end{bmatrix} \begin{bmatrix} \left(X^T X\right)^{-1}_{31} \\ \left(X^T X\right)^{-1}_{32} \\ \left(X^T X\right)^{-1}_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

After elimination we get:

$$\begin{bmatrix} 6 & -6 & 0.1182 \\ 0 & 4 & 0 \\ 0 & 0 & 0.1468 \end{bmatrix} \begin{bmatrix} \left(X^T X\right)^{-1}_{31} \\ \left(X^T X\right)^{-1}_{32} \\ \left(X^T X\right)^{-1}_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Upon backward substitution:

$$\begin{bmatrix} \left(X^T X\right)^{-1}_{13} \\ \left(X^T X\right)^{-1}_{23} \\ \left(X^T X\right)^{-1}_{33} \end{bmatrix} = \begin{bmatrix} -0.1342 \\ 0 \\ 6.812 \end{bmatrix}$$

We put all these together column by column to get:

$$\left(X^T X\right)^{-1} = \begin{bmatrix} 0.4193 & 0.25 & -0.1342 \\ 0.25 & 0.25 & 0 \\ -0.1342 & 0 & 6.812 \end{bmatrix}$$

Now that we are done with the inverse, we look towards the other parameter. *s* is calculated from:

$$s = \sqrt{\frac{1}{\nu} \Sigma_{k=1}^{N} \left(y^{[k]} - \hat{y}\left(\underline{\theta}, \underline{x}^{[k]}\right)\right)^2} \qquad (10)$$

α is specified for the appropriate confidence interval desired, here 0.05 (95% confidence intervals). ν can be calculated from:

$$\nu = N - \dim(\theta) = 6 - 3 \qquad (11)$$

The value of $T_{\nu,\alpha/2}$ is calculated from a simple piece of Matlab code or by looking it up on a chart of T-values.

Putting all of these values into the equation (9), you get 95% confidence intervals on the order of:

conf_intervals =

0.3244 0.3497

0.3208 0.3403
0.2964 0.3986

This can be accomplished with the following code:

```
% benwang_HW9_8A1.m
% Ben Wang
% HW#9 Problem #1
% due 12/9/05 9 am

% Linear Least Squares Fit

% ======= main routine benwang_HW9_8A1.m
function iflag_main = benwang_HW9_8A1();

iflag_main = 0;

%PDL> clear graphs, screen etc. general initialization
clear all; close all; clc;

Nu = [1.9676; .8986; .4261; 2.5098; 1.1521; .5520];
Re = [1; 1e-1; 1e-2; 1; 1e-1; 1e-2];
Pr = [0.73; 0.73; 0.73; 1.5; 1.5; 1.5];

X = [ones(6,1) log10(Re) log10(Pr)];

y = log10(Nu);

A = X'*X;
b = X'*y;

theta_LS = A\b
A_inv = inv(A);

% Calculate confidence intervals

alpha = 0.05;
nu = length(Nu) - length(theta_LS);

y_k = X*theta_LS;

for j = 1:6
   s_index(j) = (y_k(j) - y(j))^2;
end
sum(s_index);
s = sqrt(1/nu*sum(s_index));
t_val = tinv(1-alpha/2,nu);

for i = 1:3
   CI(i) = t_val*s*sqrt(A_inv(i,i));
   conf_intervals(i,1) = theta_LS(i) - CI(i);
   conf_intervals(i,2) = theta_LS(i) + CI(i);
```

```
end

conf_intervals

return
```

Problem 8A2

Here we are asked to do problem 8A1 with the help of the `regress` function in Matlab. This is accomplished with some very simple code, yielding:

```
theta =

    0.3370
    0.3305
    0.3475


theta_CI =

    0.3244    0.3497
    0.3208    0.3403
    0.2964    0.3986
```

You will note that you get the same values as in 8A1, with the possible exception of minor roundoff error.

```
% benwang_HW9_8A2.m
% Ben Wang
% HW#9 Problem #2
% due 12/9/05 9 am

% Linear Least Squares Fit

% ======= main routine benwang_HW9_8A2.m
function iflag_main = benwang_HW9_8A2();

iflag_main = 0;

%PDL> clear graphs, screen etc. general initialization
clear all; close all; clc;

Nu = [1.9676; .8986; .4261; 2.5098; 1.1521; .5520];
Re = [1; 1e-1; 1e-2; 1; 1e-1; 1e-2];
Pr = [0.73; 0.73; 0.73; 1.5; 1.5; 1.5];

X = [ones(6,1) log10(Re) log10(Pr)];
y = log10(Nu);
alpha = 0.05;
[theta,theta_CI] = regress(y,X,alpha)
```

```
%[theta,theta_CI,residuals,res_CI,stats]=regress(y,X,alpha)

return
```

Problem 8A3

This problem asks us to use the nlinfit function in Matlab to do regression using a nonlinear model. We now use our model that has not been linearized:

$$Nu = \alpha_0 (\mathrm{Re})^{\alpha_1} (\mathrm{Pr})^{\alpha_2} \qquad (12)$$

Matlab provides several functions that help you do regression. The first is `nlinfit`, which takes as input a design matrix of inputed predictor variables, the response variable, a function name that contains the nonlinear model, and an initial guess for the values of model parameters. It returns the least squares estimate for the parameters, the residual errors in the fit, as well as the Jacobian, which will be used to help calculate the confidence intervals.

The next function used will be `nlparci` which takes the output from `nlinfit` and uses them to calculate confidence intervals.

When we obtain parameter values and confidence intervals, with `nlinfit` and `nlparci`, we obtain:

```
theta =

   2.1857
   0.3345
   0.3400


theta_CI =

   2.1579   2.2134
   0.3252   0.3438
   0.3089   0.3710
```

If we remember now that, in comparing our nonlinear model to our linear model, we have to take the log of our pre-factor α_0. We then get values of:

```
theta =

   0.3396
   0.3345
   0.3400
```

```
theta_CI =

    0.3340    0.3451
    0.3252    0.3438
    0.3089    0.3710
```

We see that the values are slightly different, but the magnitudes of the differences are relatively negligible.

```
% benwang_HW9_8A3.m
% Ben Wang
% HW#9 Problem #3
% due 12/9/05 9 am

% Nonlinear Least Squares Fit

% ======= main routine benwang_HW9_8A3.m
function iflag_main = benwang_HW9_8A3();

iflag_main = 0;

%PDL> clear graphs, screen etc. general initialization
clear all; close all; clc;

% specify predictors and responses
Nu = [1.9676; .8986; .4261; 2.5098; 1.1521; .5520];
Re = [1; 1e-1; 1e-2; 1; 1e-1; 1e-2];
Pr = [0.73; 0.73; 0.73; 1.5; 1.5; 1.5];

X_pred = [Re Pr];
y = Nu;

%initial guess for theta
theta_0 = [0.3; 1; 1];

% calculate least squares fit using nonlinear model
[theta, residuals, Jac] = nlinfit(X_pred,y,@benwang_nlin,theta_0);
sum(residuals.^2)
% calculate confidence intervals
theta_CI = nlparci(theta,residuals,Jac);
% [y_hat,y_hat_HW] = nlpredci(@benwang_nlin,X_pred,theta,residuals,Jac);

% output theta and 95% confidence intervals
theta
theta_CI

% to compare against linear regression, we need to get alpha_0 back into
% its log form

theta(1) = log10(theta(1));
theta_CI(1,1) = log10(theta_CI(1,1));
```

```
theta_CI(1,2) = log10(theta_CI(1,2));
% output theta and 95% confidence intervals

theta
theta_CI

return

% ==== subroutine that contains model information

function y_hat = benwang_nlin(theta,X_pred)

Re = X_pred(:,1);
Pr = X_pred(:,2);

alpha_0 = theta(1);
alpha_1 = theta(2);
alpha_2 = theta(3);

% nonlinear model
y_hat = alpha_0.*(Re).^alpha_1.*(Pr).^alpha_2;

return
```