Economics 696F, Homework 6 Solutions

For the Metropolis step, note that

$$\begin{aligned} p(\gamma|y,x,y^*) &\propto p(y^*|\gamma,y,x)p(\gamma) \\ &\propto p(y^*|\gamma,x) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y_i^* - x_i'\gamma)^2\right) \\ &\propto \exp\left(-\frac{1}{2}\sum_{i=1}^{n}(y_i^* - x_i'\gamma)^2\right). \end{aligned}$$

The program `hw6.m` implements the Metropolis-within-Gibbs sampler.

Results should be similar to the results in HW5, although a poor choice of scaling factors can lead to slower convergence.

```
% hw6.m
%
% modify hw5b.m to use a Metropolis step

load hw5.mat
X = [ones(length(x),1) x];

J = 500;  % number of Gibbs draws
savestep = 1;  % save every (savestep)th draw

scal = [.1;.005];

[n,k] = size(X);

% matrix to hold gibbs draws
savedraws = zeros(ceil(J/savestep),k);

% initialize chain : could choose a different value here
beta = zeros(k,1);
Z = zeros(n,1);

% main Gibbs loop
```

```
for j=1:J,

    % save current draw for beta
    saveindex = j/savestep;
if (saveindex)==round(saveindex),
savedraws(saveindex,:) = beta';
end;

% draw for latent Z - not a super efficient approach
for i=1:n,
    stop = 0;
            while (stop==0),
Zi = X(i,:)*beta + randn;
if (Zi > 0) == y(i),
stop = 1;
end;
                Z(i) = Zi;
    end;
end;

% draw for beta via Metropolis
    beta_cand = beta + scal.*randn(k,1);
    pi_beta_old = exp(-.5*sum((Z-X*beta).^2) );
    pi_beta_cand = exp(-.5*sum((Z-X*beta_cand).^2) );
    alpha = pi_beta_cand/pi_beta_old;
    if rand<alpha,
       beta=beta_cand;
    end;
end;
```