The Econometrics of DSGE models
18/04/2013
Giuseppe Ragusa
EIEF

Problem Set #1

The file yss.csv contains data generated from the following state space models:

$$
\begin{aligned}
x_{t+1} &= g_0 + g_1 x_t + m_1 \varepsilon_{t+1}, \quad \varepsilon_t \sim N(0, \sigma_\varepsilon^2) \\
y_t &= h_0 + h_1 x_t + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2)
\end{aligned}
$$

where

$$
\begin{aligned}
g_0 &= 0 \\
g_1 &= 0.9 \\
m_1 &= 1 \\
\sigma_\varepsilon &= 0.2 \\
h_0 &= 0 \\
h_1 &= 1 \\
\sigma_\eta &= 0.3
\end{aligned}
$$

and

$$y_0 = 0, \quad x_0 = 0.$$

Let $\theta = \{g_0, g_1, m_1, \sigma_\varepsilon, h_0, h_1, \sigma_\eta\}$. The file kf_matlab.m contains code can be used to apply the Kalman's filter to $\{y_t\}_{t=1}^T$. The code reported in Listing 1. This example code call apply the Kalman's filter when $\theta = \{0, 0.9, 1, 0.2, 0, 1, 0.3\}$. The initial value are set to the unconditional mean and variance of x_t, which for this value of θ correspond to

$$
\begin{aligned}
a_0 &= E[x_t] = 0 \\
\sigma_0 &= Var[x_t] = \frac{\sigma_\epsilon^2}{1 - g_1^2} = \frac{0.2^2}{1 - 0.9^2} = 0.2105.
\end{aligned}
$$

```
% Load the yss.csv file
y = csvread('yss.csv', 1);

% Parameters
g0 = 0; g1 = .9; m1 = 1; sigmae = 0.2;
h0 = 0; h1 = 1; sigmaeta = 0.3;
% Initial values
x0 = 0;
s0 = 1/(1-g1^2)*sigmae^2;

% Call to kf_matlab
[loglik, filtered, Ptt, predicted, Pt] = ...
    kf_matlab(x0, s0, g0, g1, m1, sigmae, h0, h1,  sigmaeta, y)

>> loglik
loglik =
  -257.1391
```

The file kf_mcmc.m contains the code to estimate (a subset) of θ with a Metropolis algorithm when the likelihood is augmented with the following priors

$$\begin{aligned} g_0 &\sim N(0, 100) \\ g_1 &\sim Beta(5, 1.4) \\ \sigma_\epsilon^2 &\sim \Gamma(1, 3). \end{aligned}$$

Listing **??** gives the code of the functions defined in this file. The parameters with respect to which the posterior is calculated are $\{g_0, g_1, \sigma_\epsilon\}$. The following snippet of code show the output of the MCMC chain:

```
mcmc_chain = kf_mcmc(10000, y);

% Mean in terms of unbounded parameters
mean(mcmc_chain)

ans =
     0.0029      2.0651     -1.5910

% Mean in terms of bounded (original) parameters
mcmc_chain(:,2) = exp(mcmc_chain(:,2))./(1+exp(mcmc_chain(:,2)));
mcmc_chain(:,3) = exp(mcmc_chain(:,3));

mean(mcmc_chain)

ans =
     0.0029      0.8803      0.2050
```

1

Show how the 95% credible intervals for the three parameters that are estimated change as the priors used are

$$\begin{aligned} g_0 &\sim N(0, 100) \\ g_1 &\sim Beta(3, 2.4) \\ \sigma_\epsilon^2 &\sim \Gamma(2, 3). \end{aligned}$$

2

Modify the code to estimate all the parameters (except m_1 that is to be kept fixed at $m_1 = 1$) using Maximum Likelihood and Bayesian posterior with the following priors

$$\begin{aligned} g_0 &\sim N(0, 100) \\ g_1 &\sim Beta(3, 2.4) \\ \sigma_\epsilon^2 &\sim \Gamma(2, 3). \\ h_0 &\sim N(0, 100) \\ h_1 &\sim Beta(3, 2.4) \\ \sigma_\eta^2 &\sim \Gamma(2, 3). \end{aligned}$$

```
function [loglik, filt, Ptt, pred, Pt] = ...
             kf_matlab(x0, s0, g0, g1, m1, ...
                       sigmae, h0, h1, sigmaeta, y)
  T = size(y,1);
  % x_{t|t}
  filt   = zeros((T+1),1);
  Ptt    = zeros((T+1),0);
  Ptt(1,1)= s0;
  pred = zeros((T+1),1);
  Pt = zeros((T+1),0);

  for j=1:T
    % Predictions step
    pred(j,1) = g0+g1*filt(j,1);
    Pt(j,1) = g1*Ptt(j,1)*g1+m1^2*sigmae^2;
    % Updating step
    K = Pt(j,1)*h1*(h1^2*Pt(j,1)+sigmaeta^2)^(-1);
    filt(j+1,1) = pred(j,1) + K*(y(j,1)-h0-h1*pred(j,1));
    Ptt(j+1,1) = Pt(j,1) - K*(h1*Pt(j,1));
  end;

  mu = h0+h1*pred;
  sd = h1*Pt*h1+sigmaeta^2;
  loglik = -T*log(2*pi)/2 ...
           + sum(-log(sd(1:T))/2) ...
           -(y-mu(1:T)).^2./(2*sd(1:T)));
```

Code 1: Listings of kf_matlab.m.

```
function [chain] = kf_mcmc(sim, y)

% fix m1, h0, h1, sigmaeta
% to be estimated g0, g1, sigmae,

% Initial values for the varying parameters
% Notice this are given in terms of unbounded parameters
par0 = [0 log(.8/(1-.8)) log(0.2)];

[theta_s,FVAL,EXITFLAG,OUTPUT,GRAD,HESSIAN] = fminunc(@(x)-post(x, y)↩
    , par0);
Sigma       = inv(HESSIAN);
gamma_s     = post(theta_s, y);
for s=1:sim
    % Draw candidate
    theta_star = mvnrnd(theta_s', Sigma);
    % Construct gamma(theta*)
    gamma_star = post(theta_star, y);
    % Construct r
    r = min(exp(gamma_star-gamma_s),1);
    % Draw U
    U = rand(1);
    if(U<=r)
        theta_s = theta_star;
    end
    chain(s,:) = theta_s;
end

% This function calculate the posterior
function [post] = post(par, y)
    x0 = 0;
    % Transform the paramers
    rho = exp(par(2))/(1+exp(par(2)));
    sigmae = exp(par(3));

    s0 = sigmae^2/(1-rho^2);

    lLik = kf_matlab(x0, s0, par(1), rho, 1, sigmae, 0, 1, 0.3, y);
    % prior
    post = lLik + log(normpdf(par(1), 0, 10)) ...
            + log(betapdf(rho, 5, 1.4)) + gampdf(sigmae^2, 1, 3);
```

Code 2: Listings of kf_mcmc.m.

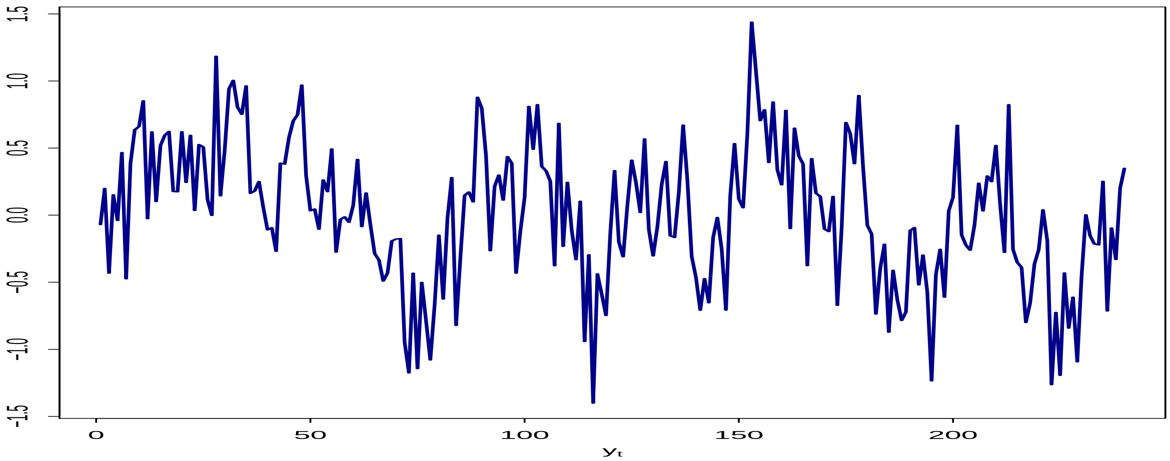

Figure 1: Plot of y variable.