The Econometrics of DSGE models
2017
Giuseppe Ragusa
EIEF

Problem Set #1

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

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

Let $\theta = \{g_0, g_1, \sigma_\varepsilon, h_0, h_1, \sigma_\eta\}$. Estimate θ under these two scenarios:

1. Flat prior on θ, that is, $p(\theta) \propto 1$. For this scenario report the MLE and the asymptotic standard errors.
2. An informative prior with marginals

$$g_0 \sim N(0, 100), \quad g_1 \sim Beta(5, 1.4), \quad \sigma_\epsilon \sim \Gamma(1, 1.5)$$
$$h_0 \sim N(0, 100), \quad h_1 \sim Beta(5, 1.4), \quad \sigma_\eta \sim \Gamma(1, 1.5)$$

For each scenario report evidence that the Metropolis Hastings algorithm has converged and report the 95% credible region.

The file yss.csv can be read by `Y = readcsv("Y.csv")`.
The following a simple implementation of a (univariate) Kalman Filter which output both the likelihood and the filtered values of x_t. This can be easiliy adapted to maximum likelihood estimation and to Bayesian estimation.

```
function kf(g0, g1, m1, sigmae, h0, h1, sigmaeta, y)
    ##
    ## x[t+1] = g0 + g1*x[t] + m*(sigmae * [t])
    ## y[t] = h0 + h1*x[t] + sigmaeta*[t]
    ##
    ## Initialize at the long run mean and variance
    ## given parameters
    x0  = g0/(1-g1)
```

```
    s0 = (m1*sigmae)^2/(1-g1^1)
    T = size(y,1)
    # x_{t|t}
    filt = Array{Float64}(T+1)
    pred  = Array{Float64}(T+1)
    Pt  = Array{Float64}(T+1)
    Ptt  = Array{Float64}(T+1)
    filt[1] = x0
    Ptt[1] = s0

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

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

    loglik, filt
end



loglik, x = kf(g0, g1, 1.0, 1.0, h0, h1, 1.0, y)
```