# Econ 623 Econometrics II

# Topic 6: Linear Gaussian State-Space Model and Kalman Filter

## 1 State Space (SS) Models

### 1.1 Why use the SS model

- Good performance (Zellner et al, 1991; Granger and Swanson, 1997)
- ARMA models assume that the data generating process (DGP) is time invariant, hence it rules out intermittent structural changes or regime shifts
- Regime shift — the switching regression model
- Parameters change over time — the time-varying parameter model.
- Both models are special cases of the SS model
- The SS model is very flexible. Many econometric models and time series models can be represented in a SS form. Examples include regression models, ARMA and ARIMA models, time-varying parameter models, unobserved component models, structural models, rational expectation models, stochastic volatility models
- The linear SS model can be easily analyzed using the Kalman filter

## 1.2 What is the linear Gaussian SS model

$$\begin{cases} y_t = A'X_t + H'\xi_t + w_t \\ \xi_t = F\xi_{t-1} + v_t \\ \begin{pmatrix} w_t \\ v_t \end{pmatrix} \overset{iid}{\sim} N\left[0, \begin{pmatrix} R & 0 \\ 0 & Q \end{pmatrix}\right] \end{cases}$$

where $y_t, X_t$ are observed, $X_t$ is exogenous, $\xi_t$ is unobserved state variable, and $A, H, F$ are matrices of parameters. Hence, there is a separate equation to describe the dynamic behavior of the state variable.

## 1.3 Examples:

1. AR(p):

   - $Y_t - \mu = \phi_1(Y_{t-1} - \mu) + \cdots + \phi_p(Y_{t-p} - \mu) + \varepsilon_t$
   - Define $\xi_t = \begin{bmatrix} Y_t - \mu \\ \vdots \\ Y_{t-p+1} - \mu \end{bmatrix}$, $F = \begin{bmatrix} \phi_1 & \cdots & \phi_{p-1} & \phi_p \\ 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}$, $v_t = \begin{bmatrix} \varepsilon_t \\ 0 \\ \vdots \\ 0 \end{bmatrix}$,

     $Q = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 1 & \cdots & 0 \end{bmatrix}$
   - $y_t = Y_t$, $X_t = 1$, $A' = \mu$, $H' = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$, $w_t = 0$, $R = 0$

2. MA(1)

   - $Y_t = \mu + \varepsilon_t + \theta\varepsilon_{t-1}$
   - Define $\xi_t = \begin{bmatrix} \varepsilon_t \\ \varepsilon_{t-1} \end{bmatrix}$, $F = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$, $v_t = \begin{bmatrix} \varepsilon_t \\ 0 \end{bmatrix}$, $Q = \begin{bmatrix} \sigma^2 & 0 \\ 0 & 0 \end{bmatrix}$
   - $y_t = Y_t$, $X_t = 1$, $A' = \mu$, $H' = \begin{bmatrix} 1 & \theta \end{bmatrix}$, $w_t = 0$, $R = 0$

# 2 Kalman filter

- Assume $I_t = \{y_1, ..., y_t, X_1, ..., X_t\}$
- The likelihood function is

  $f(y_1, ..., y_T | X_1, ..., X_T)$

  $= f(y_T | X_T, I_{T-1}) \times f(y_{T-1} | X_{T-1}, I_{T-2}) \times \cdots \times f(y_2 | X_2, I_1) \times f(y_1 | X_1)$
- Define $\widehat{\xi}_{t+1|t} = E(\xi_{t+1} | I_t)$. Call it the linear projection or optimal predictor of $\xi_{t+1}$
- Define $\widehat{\xi}_{t|t} = E(\xi_t | I_t)$. Call it the filter of $\xi_t$.
- Define $\widehat{\xi}_{t|T} = E(\xi_t | I_T)$. Call it the smoother of $\xi_t$.

- What is the distribution of $\xi_{t+1}|I_t$?

  $p(\xi_{t+1}|I_t) = \int p(\xi_{t+1}|\xi_t)p(\xi_t|I_t)d\xi_t$

- What is the distribution of $\xi_t|I_t$?

  $p(\xi_t|I_t) = \frac{p(y_t|\xi_t,X_t,I_{t-1})p(\xi_t|X_t,I_{t-1})}{p(y_t|X_t,I_{t-1})} = \frac{p(y_t|\xi_t,X_t)p(\xi_t|I_{t-1})}{p(y_t|X_t,I_{t-1})}$

  where $p(y_t|X_t, I_{t-1}) = \int p(y_t|\xi_t, X_t, I_{t-1})p(\xi_t|X_t, I_{t-1})d\xi_t = \int p(y_t|\xi_t, X_t)p(\xi_t|I_{t-1})d\xi_t$

- What is the distribution of $\xi_t|I_T$?

  $p(\xi_t|I_T) = \int p(\xi_t, \xi_{t+1}|I_T)d\xi_{t+1} = \int p(\xi_t|\xi_{t+1}, I_T)p(\xi_{t+1}|I_T)d\xi_{t+1}$

  $\int p(\xi_t|\xi_{t+1}, I_t)p(\xi_{t+1}|I_T)d\xi_{t+1} = p(\xi_t|I_t) \int \frac{p(\xi_{t+1}|\xi_t)p(\xi_{t+1}|I_T)}{p(\xi_{t+1}|I_t)}d\xi_{t+1}$

- In linear Gaussian state-space models, all these densities are Gaussian and hence uniquely determined by the first two moments.

- Kalman filter calculates the linear projections recursively, generating $\widehat{\xi}_{1|0}, \widehat{\xi}_{2|1}, ...$ and the associated mean square errors of these forecasts, defined by $E(\xi_{t+1} - \widehat{\xi}_{t+1|t})^2 \equiv \Sigma_{t+1|t}$. It also calculates the filter, the smoother and $p(y_1|X_1), p(y_2|X_2, y_1), ....$

- Define $y_{t|t-1} = y_t|X_t, I_{t-1}$ and $\widehat{y}_{t|t-1} = E(y_t|X_t, I_{t-1})$

- Multi-step procedure

1. Initialization: $\xi_{1|0} \sim N(E(\xi_1), Var(\xi_1))$. So

$$\begin{cases} \widehat{\xi}_{1|0} = E(\xi_1) \\ \Sigma_{1|0} = E[\xi_1 - E(\xi_1)]^2 = Var(\xi_1) \end{cases} .$$

These are the unconditional mean and the unconditional variance. And $\xi_1 | X_1 = \xi_{1|0}$.

2. $y_1|X_1 = A'X_1 + H'\xi_1|X_1 + w_1|X_1 = A'X_1 + H'\xi_{1|0} + w_1$. So

$$\begin{cases} \widehat{y}_{1|0} = E(y_1|X_1) = A'X_1 + H'\widehat{\xi}_{1|0} \\ E[(y_1 - \widehat{y}_{1|0})^2] = H'\Sigma_{1|0}H + R \end{cases} .$$

3. $\widehat{\xi}_{1|1} = E(\xi_1|y_1, X_1) = E(\xi_1|X_1) + E[(\xi_1 - \widehat{\xi}_{1|0})(y_1 - \widehat{y}_{1|0})]$

   $\times\{E[(y_1 - \widehat{y}_{1|0})^2]\}^{-1} \times (y_1 - \widehat{y}_{1|0})$

   $= \xi_1|X_1 + \Sigma_{1|0}H(H'\Sigma_{1|0}H + R)^{-1}(y_1 - A'X_1 - H'\widehat{\xi}_{1|0})$

   The associated MSE is $\Sigma_{1|1} = \Sigma_{1|0} - \Sigma_{1|0}H(H'\Sigma_{1|0}H + R)^{-1}H'\Sigma_{1|0}$

   This step is called updating.

4. $\widehat{\xi}_{2|1} = E(\xi_2|y_1, X_1) = FE(\xi_1|y_1, X_1) = F\widehat{\xi}_{1|1}$

   The associated MSE is $\Sigma_{2|1} = F\Sigma_{1|1}F' + Q$

   This step is called in-sample prediction

5. Repeat Steps 2-4 for $t = 1, ..., T$.

6. Calculate the following recursive expressions backward to get the smoother
   $$\begin{cases} \widehat{\xi}_{t|T} = \widehat{\xi}_{t|t} + J_t(\xi_{t+1} - \widehat{\xi}_{t+1|t}) \\ \Sigma_{t|T} = \Sigma_{t|t} + J_t(\Sigma_{t+1|T} - \Sigma_{t+1|t})J_t' \end{cases},$$
   where $J_t = \Sigma_{t|t}F'\Sigma_{t+1|t}^{-1}$

7. Calculate $\begin{cases} \widehat{\xi}_{T+h|T} = F^h\widehat{E}(\xi_T|I_T) = F^h\widehat{\xi}_{T|T} \\ \widehat{y}_{T+h|T} = A'x_{T+h} + H'F^h\widehat{\xi}_{T|T} (\text{if } x \text{ is deterministic}) \end{cases}.$

   This step is called out-of-sample forecasting

- Summary

  1. Initialization:
  $$\begin{cases} \widehat{\xi}_{1|0} = E(\xi_1) \\ \Sigma_{1|0} = E[\xi_1 - E(\xi_1)]^2 = Var(\xi_1) \end{cases}.$$
  2. Sequential updating
  $$\begin{cases} \widehat{\xi}_{t|t} = \widehat{\xi}_{t|t-1} + \Sigma_{t|t-1}H(H'\Sigma_{t|t-1}H + R)^{-1}(y_t - A'x_t - H'\widehat{\xi}_{t|t-1}) \\ \Sigma_{t|t} = \Sigma_{t|t-1} - \Sigma_{t|t-1}H(H'\Sigma_{t|t-1}H + R)^{-1}H'\Sigma_{t|t-1} \end{cases}.$$
  3. In-sample sequential prediction
  $$\begin{cases} \widehat{\xi}_{t+1|t} = F\widehat{\xi}_{t|t} = F\widehat{\xi}_{t|t-1} + \\ \qquad\qquad F\Sigma_{t|t-1}H(H'\Sigma_{t|t-1}H + R)^{-1}(y_t - A'x_t - H'\widehat{\xi}_{t|t-1}) \\ \Sigma_{t+1|t} = F\Sigma_{t|t}F' + Q \end{cases}.$$
  $$\begin{cases} \widehat{y}_{t+1|t} = A'x_{t+1} + H'\widehat{\xi}_{t+1|t} \\ E[(y_{t+1} - \widehat{y}_{t+1|t})(y_{t+1} - \widehat{y}_{t+1|t})'] = H'\Sigma_{t+1|t}H + R \end{cases}.$$
  4. Smoothing
  $$\begin{cases} \widehat{\xi}_{t|T} = \widehat{\xi}_{t|t} + J_t(\widehat{\xi}_{t+1|T} - \widehat{\xi}_{t+1|t}) \\ \Sigma_{t|T} = \Sigma_{t|t} + J_t(\Sigma_{t+1|T} - \Sigma_{t+1|t})J_t' \end{cases}.$$
  5. Out-of-sample forecasting
  $$\begin{cases} \widehat{\xi}_{T+h|T} = F^h\widehat{E}(\xi_T|I_T) = F^h\widehat{\xi}_{T|T} \\ \widehat{y}_{T+h|T} = A'x_{T+h} + H'F^h\widehat{\xi}_{T|T}(\text{if } x \text{ is deterministic}) \end{cases}.$$

- First three steps are used to obtain the likelihood function

- Last step is used to generate optimal forecasts

# MATLAB Program for ML Estimation of A State-Space Model

```
function z=ML_LinearSS(nrep,theta)
%Use ML to estimate the parameters in simple linear Gaussian SS model
%y(t)=x(t)+u(t), x(t)=theta*x(t-1)+v(t)
%u~N(0,\sigma_u^2), v~N(0,1), u and v are uncorrelated
global nob;
global sigmau2;
global y;
nob=100;%user has to provide the number of observations
sigmau2=1;%user has to provide the value for this parameter
options = optimset('Display','Iter','TolFun',1e-6);
randn('seed',12345);
u=normrnd(0,1,nrep,nob)*sqrt(sigmau2);
for i=1:nrep
    randn('seed',i);
    v=normrnd(0,1,nob,1);
    v(1)=0;
    x=filter(1, [1 -theta], v);
    zz(i,1)=sum(x(2:nob).*x(1:nob-1))/sum(x(1:nob-1).^2);
    y=x+u(i,:)';
    xyz= fminsearch(@llss,[zz(i,1) sigmau2 1],options)
    zz(i,2)=xyz(1);
end
xy=[theta mean(zz) var(zz)]
return;
%
function x=llss(theta1)
global nob;
global sigmau2;
global y;
theta=theta1(1);
sigmau2=theta1(2);
sigmav2=theta1(3);
xii(1)=0;%initialization
sigi(1)=sigmav2/(1-theta^2);%initialization
myinv=sigi(1) + sigmau2;%producing H'PH+R in Eq 13.4.1 in Hamilton
myarg=y(1) - xii(1);
lnl(1)=log(myinv)+myarg^2/myinv;
for i =2:nob
    gain=theta * sigi(i - 1) /myinv;%gain matrix
    xii(i)=theta * xii(i - 1) + gain*myarg;%forecast
    sigi(i)=theta^2 * sigi(i - 1) - gain*theta*sigi(i - 1)+sigmav2;%forecast
    myinv=sigi(i) + sigmau2;
    myarg=y(i) - xii(i);
    lnl(i)=log(myinv)+myarg^2/myinv;
end
x=(sum(lnl)+nob*log(2*pi))*0.5;
return
```