Lecture 5: Statistical Model in Macroeconomics (II)

Instructor: Yucheng Yang *Scribe: Yuqing Ding, Zeru Jiang, Yiming Qiao, Sun Yang*

1 Vector Autoregressive Model with Distributional Inputs

1.1 Model Setup

- Target: add micro variables (distributional variables) into VAR and SVAR.
- Input variables:
  - Z_t : macroeconomic aggregate variables.
  - $\ell_t(x) = \log P_t(x)$: log density function for one-dimensional variable $x_{i,t}$ (could be extended to multivariate case later).
- VAR $\longrightarrow$ Functional VAR: Decompose Z_t and ℓ_t into a deterministic component $(Z_*, \ell_*(x))$ and fluctuations $Z_t = Z_* + \tilde{Z}_t, \ell_t = \ell_* + \tilde{\ell}_t$. $(\tilde{Z}_t, \tilde{\ell}_t(x))$ evolve according to linear **functional VAR**:

$$\begin{aligned} \tilde{Z}_t &= B_{zz}\tilde{Z}_{t-1} + \int B_{zl}(\tilde{x})\tilde{\ell}_{t-1}d\tilde{x} + u_{z,t} \\ \tilde{\ell}_t(x) &= B_{lz}(x)\tilde{Z}_{t-1} + \int B_{ll}(x,\tilde{x})\ell_{t-1}(\tilde{x})d\tilde{x} + u_{l,t}(x). \end{aligned} \tag{1}$$

  with kernel functions $B_{zl}(\tilde{x})$ and $B_{ll}(x, \tilde{x})$.
- Dimension reduction: how to reduce the dimension of $\tilde{\ell}_t(x)$?
  1. Moments. First moment, second moment, etc. Generalized moments also possible.
  2. Nonparametric estimation. Eg. Sieve approximation; autoencoder (popular in many real world applications).
  3. Dimension reduction with economic theory. E.g. mass of certain type of agents.
- Sieve approximations (Chang, Chen and Schorfheide, 2018):

$$\tilde{\ell}_t(x) \approx \tilde{\ell}_t^K = \sum_{k=0}^{K} \tilde{\alpha}_{k,t}\zeta_k(x) = [\zeta_0(x), \zeta_1(x), ..., \zeta_K(x)] \begin{bmatrix} \tilde{\alpha}_{0,t} \\ \vdots \\ \tilde{\alpha}_{K,t} \end{bmatrix} = \zeta_t(x)\tilde{\alpha}_t \tag{2}$$

with basis functions $\{\zeta_k(x)\}_{k=0}^K$. Then we can get the new VAR formulation

$$\text{fVAR Equation } (1) \Rightarrow \begin{bmatrix} \tilde{Z}_t \\ \tilde{\alpha}_t \end{bmatrix} = \begin{bmatrix} B_{zz} & B_{zl}C_\alpha \\ B_{lz} & B_{ll}C_\alpha \end{bmatrix} \begin{bmatrix} \tilde{Z}_{t-1} \\ \tilde{\alpha}_{t-1} \end{bmatrix} + \begin{bmatrix} u_{z,t} \\ u_{\alpha,t} \end{bmatrix} \tag{3}$$

Where

$$C_\alpha = \int \xi'(\tilde{x})\zeta(\tilde{x})d\tilde{x}. \tag{4}$$

with another sequence of basis functions $\{\xi_j(x)\}_{j=0}^J$.

- Estimation: In equation (3), we can calculate $\tilde{Z}_t, \tilde{\alpha}_t, C_\alpha$ with data and basis functions we choose. Then we may estimate Equation (3) directly using Bayesian methods, but may also estimate a state space formulation like Chang et al. (2018) if we consider measurement errors of the observed distribution.

1.2 Implementation

- Basis functions: splines. A spline is a piecewise polynomial functions with knots $x_s, s = 1, 2, ..., S$:

$$\begin{aligned} Spl(m,S) &= \sum_{k=0}^{m} a_k(x^k \mathbb{1}\{x \le x_S\} + x_S^k \mathbb{1}\{x > x_S\}) \\ &+ \sum_{s=1}^{S-1} b_s([max\{x - x_s, 0\}]^m \mathbb{1}\{x \le x_S\} + (x_S - x_s)^m \mathbb{1}\{x > x_S\}) \\ &+ \sum_{t=1}^{m} C_k[max\{x - x_S, 0\}]^m \end{aligned} \tag{5}$$

$m = 3, a_2 = a_1 = 0, c_2 = c_3 = 0 \Rightarrow$ tails of Laplace deasity
And we can obtain that

$$\begin{aligned} \zeta_0(x) &= 1 \\ \zeta_1(x) &= (x\mathbb{1}\{x \le x_S\} + x_S\mathbb{1}\{x > x_S\}) \\ \zeta_2(x) &= ([max\{x - x_1, 0\}^3 \mathbb{1}\{x \le x_S\} + (x_S - x_1)^3 \mathbb{1}\{x > x_S\}) \\ &\vdots \\ \zeta_S(x) &= (max\{x - x_{S-1}, 0\}^3 \mathbb{1}\{x \le x_S\} + (x_S - x_{S-1})^3 \mathbb{1}\{x > x_S\}) \\ \zeta_{S+1}(x) &= max\{x - x_S, 0\} \end{aligned} \tag{6}$$

- Sieve Coefficients: we obtain $\hat{\alpha^0_{k,t}}$ with sieve approximation for the density function at each time t. Chang et al. (2018) also apply seasonal adjustment and compression (PCA) on $\hat{\alpha^0_{k,t}}$ to get rid of seasonality and collinearity. The new functional VAR after the seasonal adjustment and compression operations share the same structure as equation (3) which could be estimated using Bayesian method.

- All the operations so far are reversible so that we can go back to the original variables after computing impulse responses and other things with the functional VAR.
- Bayesian estimation of functional VAR like equation (3).
- Recover densities from α's.

1.3 Empirical Results

The model can be used to look into:

- VAR-type questions:
    1. How distributional dynamics affect aggregate dynamics?
    2. How aggregate dynamics affect distributional dynamics?
- SVAR-type questions (with further identification assumptions):
    1. How distributional shocks affect aggregate dynamics?
    2. How aggregate shocks affect distributional dynamics?

2 State Space Model with Distributional Inputs

Similar to vector autoregressive models, state space models can also handle micro distributional inputs. We introduce the framework proposed by Liu and Plagborg-Moller (2019) in this section.

2.1 Model Setup and Key Assumptions

The structure of the state space model is visualized in Figure 1.

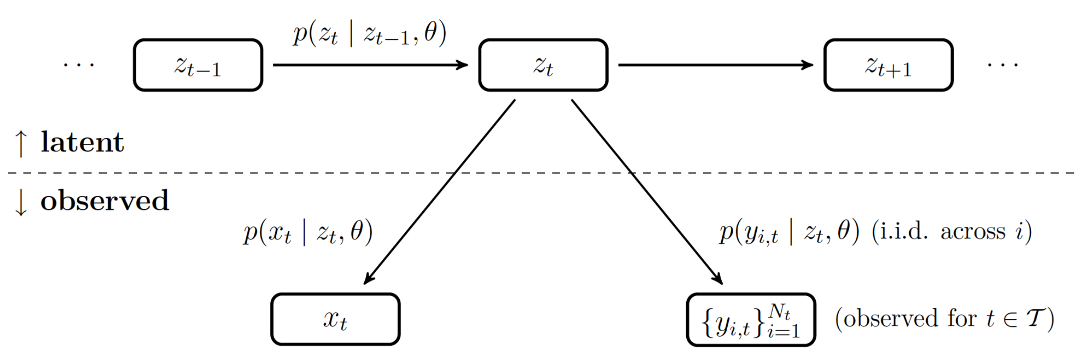

Figure 1: Diagram of state space model with both micro and macro observables.

- Observables: macro variables $x_t \sim p(x_t|z_t, \theta)$, micro distributional variables $y_{i,t} \sim p(y_{i,t}|z_t, \theta)$ and is i.i.d. across different i.
- Latent states: $z_t \sim p(z_t|z_{t-1})$. The latent variables could be observed or unobserved macro variables, could also be distributional variables (e.g. the whole distribution). However, the latent state is shared among all the individuals, and cannot be identified as $z_{i,t}$.
- Here they impose assumptions that $y_{i,t}$ is independent of $y_{i,t-1}$ given macro latent state z_t.

2.2 Likelihood and Estimation

The likelihood can be written as:

$$p(x, y|\theta) = \underbrace{p(x|\theta)}_{\text{Macro Part}} \underbrace{p(y|x, \theta)}_{\text{Distributional Part}} \tag{7}$$

The **macro part** could be obtained via standard state space likelihood formulation:

$$p(x|\theta) = \prod_{t=1}^{T} p(x_t|\mathcal{X}_{t-1}, \theta) = \prod_{t=1}^{T} p(x_t|z_t, \theta)p(z_t|z_{t-1})p(z_{t-1}|\mathcal{X}_{t-1}, \theta) \tag{8}$$

where $\mathcal{X}_t = \{x_\tau\}_{\tau=1}^{t}$ and $p(z_{t-1}|\mathcal{X}_{t-1}, \theta)$ is obtained from Kalman filter with initial assumption on $p(z_0|\mathcal{X}_0, \theta) = p(z_0|\theta)$.

The **distributional part** could be written as

$$\begin{aligned} p(y|x, \theta) &= \int p(y|z, x, \theta)p(z|x, \theta)dz = \int p(y|z, \theta)p(z|x, \theta)dz \\ &= \int \prod_{t=1}^{T} \prod_{i=1}^{N_t} p(y_{it}|z_t, \theta)p(z|x, \theta)dz \end{aligned} \tag{9}$$

If we can sample draws $\mathbf{z}^{(j)} \equiv \left\{ z_t^{(j)} \right\}_{1 \le t \le T}, j = 1, \ldots, J$ from the density

$$p(z|x, \theta) = \prod_{t=1}^{T} p(z_t|\mathcal{X}_t, \theta)$$

which can be obtained from the Kalman smoother, then we can approximate the distributional part of the likelihood with

$$p(y|x, \theta) \approx \frac{1}{J} \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{i=1}^{N_t} p(y_{it}|z_t = z_t^{(j)}, \theta) \tag{10}$$

3 Recurrent Neural Network and LSTM Network

One of the most important formulation of nonlinear state space models is the recurrent neural networks (RNN). We introduce RNN and its variant long short-term memory (LSTM) networks in this section.

3.1 Recurrent neural network (RNN)

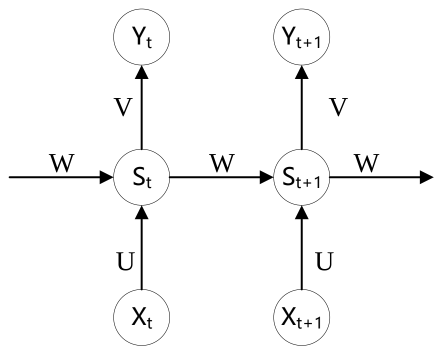

Figure 2: An unrolled recurrent neural network.

Similar to linear state space model, RNN has hidden states S_t, which affect the output Y_t together with input variables X_t as illustrated in Figure 2.

$$\begin{aligned} S_t &= \sigma(WS_{t-1} + UX_t + b) \\ Y_t &= VS_t \end{aligned}$$

where

- $X_t \in \mathcal{R}^{k\times 1}$: input variables.
- $S_t \in \mathcal{R}^{n\times 1}$: state variables.
- $Y_t \in \mathcal{R}^{m\times 1}$: output variables.
- $W \in \mathcal{R}^{n\times n}, U \in \mathcal{R}^{k\times n}, V \in \mathcal{R}^{n\times m}, b \in \mathcal{R}^{n\times 1}$: parameters.

Remark: With constant transition matrix over time, RNN suffers from so-called “gradient vanishing” problem, so it could not deal with problems with “long-term dependencies”. LSTM is proposed to handle this problem.

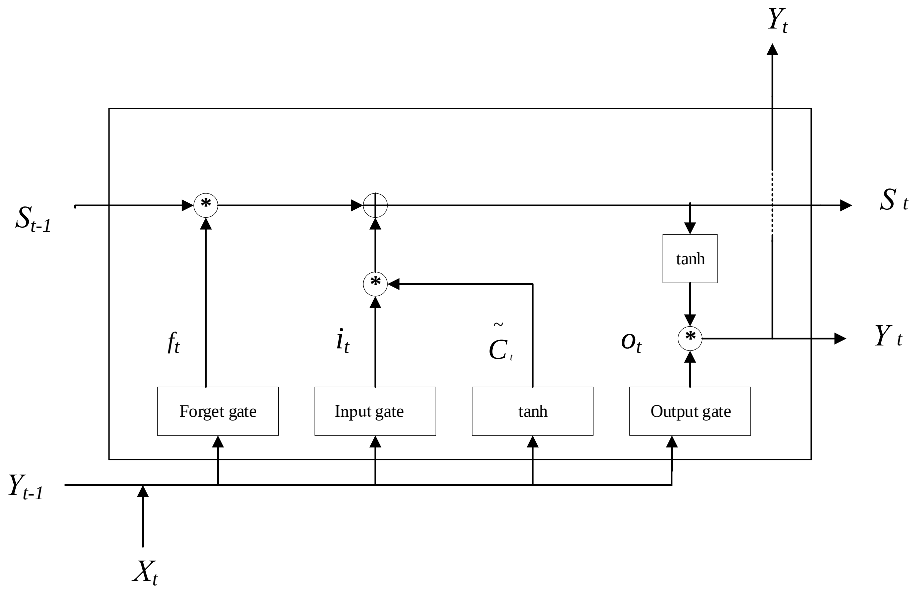

Figure 3: A LSTM cell

3.2 Long Short-term Memory (LSTM)

Figure 3 shows the detailed structure of a LSTM cell. The core of LSTM is the design of three gates: *forget gate*, *input gate* and *output gate*.

- **Forget gate.** It decides what information LSTM is going to throw away from the cell state.

$$f_t = \sigma(W_f \cdot [Y_{t-1}, X_t] + b_f)$$

where σ indicates the sigmoid activation function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- **Input gate.** It decides what new information LSTM is going to store in the cell state.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [Y_{t-1}, X_t] + b_i) \\ \tilde{C}_t &= \tanh(W_c \cdot [Y_{t-1}, X_t] + b_i) \end{aligned}$$

where $\tilde{C}_t$ is a vector of new candidates values, which will be used to update cell state.

- **Update cell state.** S_{t-1} denotes the old cell state, and then update it into new cell state S_t by:

$$S_t = f_t * S_{t-1} + i_t * \hat{S}_t$$

- **Output gate.** It decides what LSTM is going to output.

$$\begin{aligned} o_t &= \sigma(W_o \cdot [Y_{t-1}, X_t] + b_o) \\ Y_t &= o_t * \tanh(S_t) \end{aligned}$$

  LSTM put the cell state through $\tanh$ ($\tan(S_t) \in [-1, 1]$) and multiply it by the output of the sigmoid gate, then get the output Y_t.

Reference

Chang, M., Chen, X. and Schorfheide, F., 2018. Heterogeneity and Aggregate Fluctuations. Manuscript, University of Pennsylvania.

Liu, L. and Plagborg-Møller, M., 2019. Full-Information Estimation of Heterogeneous Agent Models Using Macro and Micro Data.

Christopher Olah. 2015. Understanding LSTM Networks. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

Karpathy, A., 2015. The unreasonable effectiveness of recurrent neural networks. `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`.