DSGE-SVt: An Econometric Toolkit for High-Dimensional DSGE Models with SV and t Errors

Siddhartha Chib
Washington University in St. Louis

Minchul Shin
Philadelphia Fed

Fei Tan
Saint Louis University

MEC 670 Seminar in Econometrics

Washington University in St. Louis

January 5, 2023

Introduction

- DYNARE has played a large role in the practical fitting of DSGE models
- However, it cannot handle *high-dimensional* DSGE models or models with
  - Student-t shocks, e.g., Chib & Ramamurthy (2014)
  - stochastic volatility, e.g., Justiniano & Primiceri (2008)
- We provide a user-friendly MATLAB toolbox for such models that contains
  - training sample priors
  - efficient sampling of parameters by the TaRB-MH algorithm of Chib & Ramamurthy (2010)
  - fast computation of the marginal likelihood by the method of Chib (1995) and Chib & Jeliazkov (2001)
  - post-estimation tools, e.g., point and density forecasts

Illustration

- Dynamic IS relation

$$\hat{y}_t = \mathbb{E}_t \hat{y}_{t+1} + \hat{g}_t - \mathbb{E}_t \hat{g}_{t+1} - \frac{1}{\tau}(\hat{R}_t - \mathbb{E}_t \hat{\pi}_{t+1} - \mathbb{E}_t \hat{z}_{t+1})$$

- New Keynesian Phillips curve

$$\hat{\pi}_t = \beta \mathbb{E}_t \hat{\pi}_{t+1} + \kappa(\hat{y}_t - \hat{g}_t)$$

- Monetary policy rule

$$\hat{R}_t = \rho_R \hat{R}_{t-1} + (1 - \rho_R)\psi_1 \hat{\pi}_t + (1 - \rho_R)\psi_2(\hat{y}_t - \hat{g}_t) + \epsilon_{R,t}$$

- Exogenous shocks

$$\hat{g}_t = \rho_G \hat{g}_{t-1} + \epsilon_{G,t}, \quad \hat{z}_t = \rho_Z \hat{z}_{t-1} + \epsilon_{Z,t}$$

- t-innovation with SV: for $s \in \{R, G, Z\}$

$$\epsilon_{s,t} \sim t_\nu(0, e^{h_{s,t}}), \quad h_{s,t} = (1 - \phi_s)\mu_s + \phi_s h_{s,t-1} + \eta_{s,t}, \quad \eta_{s,t} \sim N(0, \omega_s^2)$$

State Space Form

- We rely on Sims' (2001) method to solve for the solution of the structural model, for *each* value of θ
- After (log) linearizing around the steady state, the structural model is expressed in canonical form as

$$\Gamma_0(\theta)x_t = \Gamma_1(\theta)x_{t-1} + \Psi\epsilon_t + \Pi\eta_t$$

- Applying Sims' method, the unique, bounded solution takes the form

$$x_t = G(\theta)x_{t-1} + M(\theta)\epsilon_t$$

where $G(\theta)$ and $M(\theta)$ are non-linear unspecified functions of θ determined by the solve step

State Space Form

- Model completed by the measurement equations

$$\begin{pmatrix} \mathsf{YGR}_t \\ \mathsf{INF}_t \\ \mathsf{INT}_t \end{pmatrix} = \begin{pmatrix} \gamma^{(Q)} \\ \pi^{(A)} \\ \pi^{(A)} + r^{(A)} + 4\gamma^{(Q)} \end{pmatrix} + \begin{pmatrix} \hat{y}_t - \hat{y}_{t-1} + \hat{z}_t \\ 4\hat{\pi}_t \\ 4\hat{R}_t \end{pmatrix}$$

- We show results for an extended version of this model that has 51 parameters, 21 variables, 8 shocks, 8 observables, and 1,494 non-Gaussian and nonlinear latent variables

Example

- Set MATLAB directory to the 'DSGE-SVt' folder
- Specify the model, data, and save directories

```
%% Housekeeping
clear
close all
clc

%% User search path & mex files
modpath = ['user' filesep 'ltw17'];
datpath = ['user' filesep 'ltw17' filesep '
    data.txt'];
savepath = ['user' filesep 'ltw17'];
ltw17 = tarb(@tarb_spec,[],'modpath',modpath,'
    datpath',datpath,'savepath',savepath);

OneFileToMexThemAll
```

Training Sample Prior

- In high dimensions, formulating an appropriate prior is difficult due to the complex mapping from θ to $G(\theta)$ and $M(\theta)$
- Standard choices often produce prior-sample conflict
- We supply two ways of dealing with this: training sample priors and Student-t family of distributions for location-type parameters
- A sampling the prior function is available to calculate the implied distribution of the outcomes

```
%% Sampling the prior
npd = 10000;          % number of prior draws
T = 200;              % number of periods
sdof = 5;             % shock degrees of freedom
ltw17 = tarb(@tarb_spec,ltw17,'npd',npd,'
    periods',T,'sdof',sdof);
tarb(@sample_prior,ltw17);
```

Estimation

- The centerpiece of the estimation procedure is the TaRB-MH algorithm of Chib & Ramamurthy (2010)
- Used to draw samples of $(\theta, z_{1:T})$ from

$$\pi(\theta, z_{1:T}|y_{1:T}) \;\propto\; f(y_{1:T}, z_{1:T}|\theta) \cdot \pi(\theta) \cdot \mathbf{1}\{\theta \in \Theta_D\}$$

- TaRB-MH is coded up in DYNARE, but the implementation there is somehow not efficient

TaRB-MH MCMC

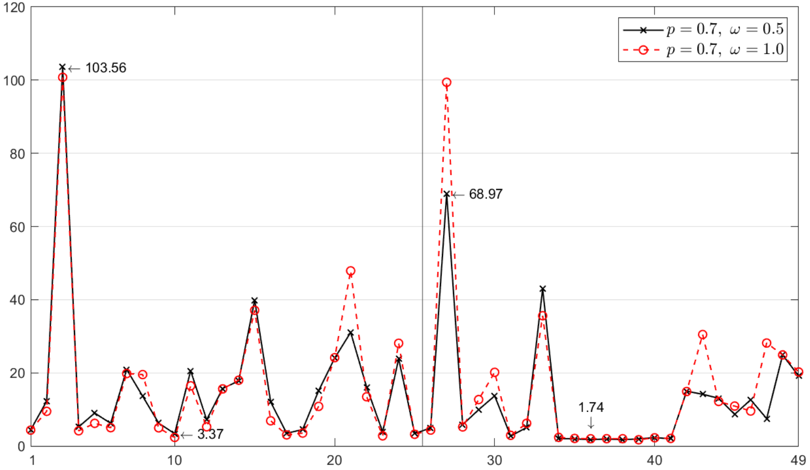
- Two hallmarks of TaRB-MH
  - randomize the number and components of blocks
  - tailor the proposal density to the posterior location and curvature, i.e., $q_b(\theta_b|y_{1:T}, z_{1:T}, \theta_{-b}) = t_v(\theta_b|\hat{\theta}_b, \hat{V}_b)$, $b = 1, \ldots, B$, using simulated annealing and Chris Sims' `csminwel`
- The toolbox provides a fast implementation of these steps by
  - randomly tailoring proposal every few iterations
  - MEX loop-intensive functions from C/C++ code
  - `csminwel` to get $\hat{V}_b$ as output
- Although intensive it all happens seamlessly

Example

- Set up the TaRB-MH algorithm
- The estimation results are stored in the MATLAB data file `tarb_full.mat`

```
%% TaRB-MH (full run)
p = 0.7;                  % blocking probability
w = 0.5;                  % tailoring frequency
M = 11000;                % number of draws
N = 1000;                 % number of burn-in
ltw17 = tarb(@tarb_spec,ltw17,'prob',p,'freq',
    w,'M',M,'N',N);
tarb(@sample_post,ltw17);
```

Inefficiency Factor



Model Comparison

- Marginal likelihood is computed by a fast implementation of the Chib & Jeliazkov (2001) estimator
- This estimator is based on the identity of Chib (1995)

$$m(y_{1:T}) = \frac{f(y_{1:T}|\theta^*)\pi(\theta^*)}{\pi(\theta^*|y_{1:T})}, \quad \forall\theta^*$$

Model Comparison

- With multiple block sampling, the posterior ordinate is estimated from the decomposition

$$\pi(\theta^*|y_{1:T}) = \pi(\theta_1^*|y_{1:T})\pi(\theta_2^*|y_{1:T},\theta_1^*)\cdots\pi(\theta_B^*|y_{1:T},\theta_1^*,\ldots,\theta_{B-1}^*)$$

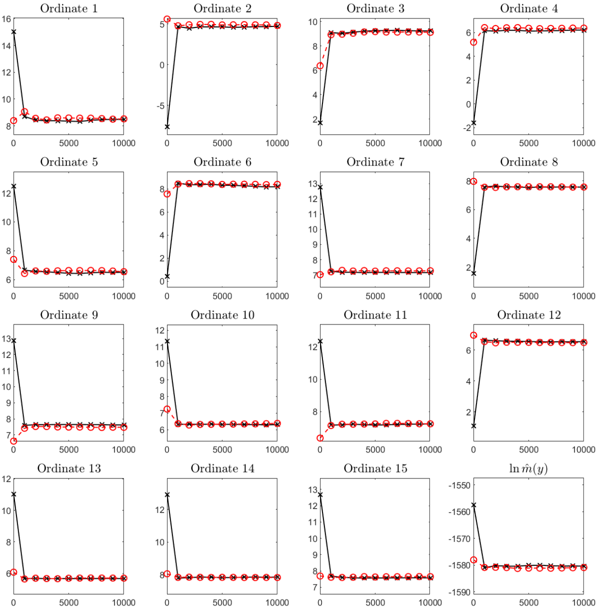
  using several reduced TaRB-MH runs
- Key point: all reduced runs can be parallelized for the cost of only one reduced run, regardless of the number of blocks
- This speeds up the calculation enormously

Example

- Set up the TaRB-MH reduced run
- The estimation results are stored in the MATLAB data file `tarb_reduce.mat`

```
%% TaRB-MH (reduced run)
B = 7;                  % number of blocks
ltw17 = tarb(@tarb_spec,ltw17,'blocks',B);
tarb(@marg_lik,ltw17);
```

Marginal Likelihood



Simulation Evidence

Table 1: Chib-Jeliazkov estimator

DGP 1: regime-M with $\nu = 15$			DGP 2: regime-F with $\phi = 0.5$		
ν	M	F	ϕ	M	F
30 (light)	4	0	0.1 (weak)	0	9
15 (fat)	15	0	0.5 (moderate)	0	10
5 (heavy)	1	0	0.9 (strong)	0	1

NOTES: Number of picks for each model specification.

Table 2: Modified harmonic mean estimator

DGP 1: regime-M with $\nu = 15$			DGP 2: regime-F with $\phi = 0.5$		
ν	M	F	ϕ	M	F
30 (light)	0	0	0.1 (weak)	0	0
15 (fat)	0	0	0.5 (moderate)	0	0
5 (heavy)	20	0	0.9 (strong)	0	20

Post-Estimation Tools

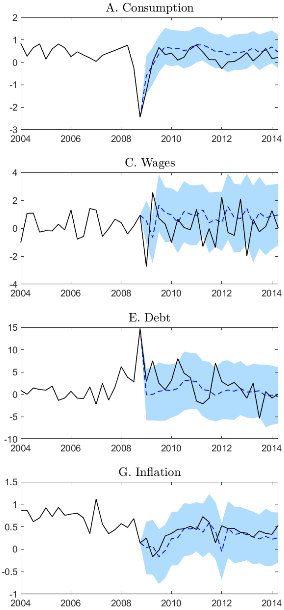
- Toolbox includes post-estimation tools, e.g., functions for conducting impulse response, variance decomposition, and prediction
- For instance, suppose we want the one-quarter-ahead prediction density

$$p(y_{T+1}|y_{1:T}) = \int p(y_{T+1}|y_{1:T}, \theta, z_{T+1}) \cdot \pi(\theta, z_{T+1}|y_{1:T}) d(\theta, z_{1:T})$$

- This is available through the TaRB-MH specification

```
%% Out-of-sample forecast
head = 50;          % training sample size
tail = 22;          % forecasting sample size
h = 1;              % forecasting horizon
ltw17 = tarb(@tarb_spec,ltw17,'datrange',[head
    tail],'pred',h);
```

Out-of-Sample Forecast



Outroduction

- Going beyond DYNARE for Next-Gen DSGE models
- Efficient and fast estimation via TaRB-MH and parallel computing
- Toolkit available at github.com/econdojo/dsge-svt
- Readily applied, e.g., open economy DSGE as in Lin (2021), post COVID-19 DSGE forecasting as in Chib, Drautzburg, Shin & Tan (in progress)

References

- Chib & Ramamurthy (2010), "Tailored Randomized Block MCMC Methods with Application to DSGE Models," *Journal of Econometrics*
- Chib, Shin, & Tan (2021), "DSGE-SVt: An Econometric Toolkit for High-Dimensional DSGE Models with SV and t Errors," *Computational Economics*
- Sims (2001), "Solving Linear Rational Expectations Models," *Computational Economics*