Method of Moments Estimation in Dynare

Willi Mutschler

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Dynare Team

Motivation

Dynare 4.7 features a new *Method of Moments* toolbox that provides functionality to estimate parameters by

- Simulated Method of Moments (SMM) up to any perturbation approximation order (with or without *pruning*)

- Generalized Method of Moments (GMM) up to 3rd-order *pruned* perturbation approximation

Toolbox is inspired by replication codes accompanied to Andreasen et al. (2018), Born and Pfeifer (2014), and Mutschler (2018)

Computing moments in Dynare

Dynare's model framework

General DSGE model

$$E_t f\left(y_{t+1}, y_t, y_{t-1}, u_t | \theta\right) = 0$$

$$y_t = g(x_{t-1}, u_t | \theta)$$

$$u_t = \sigma \eta_t \quad \text{with } \eta_t \sim N(0,\Sigma)$$

Perturbation solution: Taylor-approximation around the non-stochastic steady-state (where y_t are all endogenous and x_t are states):

$$y_t = \bar{y} + g_x(x_{t-1} - \bar{x}) + g_u u_t + \frac{1}{2}\left[g_{xx}(x_{t-1} - \bar{x}) \otimes (x_{t-1} - \bar{x}) + 2g_{xu}(x_{t-1} \otimes u_t) + g_{uu}(u_t \otimes u_t) + g_{\sigma\sigma}\sigma^2\right] + \frac{1}{6}[\ldots] + \ldots$$

How to compute moments?

- Via simulation (periods option): Use approximated policy function to simulate data with large sample size and compute the empirical mean, covariance, autocovariance, skewness, kurtosis, ...
- Use theoretical expressions based on state space representation

First-order approximation

$$y_t = \bar{y} + g_x(x_{t-1} - \bar{x}) + g_u u_t$$
$$x_t = \bar{x} + h_x(x_{t-1} - \bar{x}) + h_u u_t$$

The unconditional first and second moments are (given $E[u_t] = 0$ and $E[u_t^2] = \Sigma_u = \sigma^2\Sigma$)

$$E[y_t] \equiv \mu_y = \bar{y}, \quad E(x_t) \equiv \mu_x = \bar{x}$$

$$\Sigma_y(0) \equiv E[(y_t - \bar{y})(y_t - \bar{y}')] = g_x \Sigma_x(0) g_x' + g_u \Sigma_u g_u'$$

$$\Sigma_x(0) \equiv E[(x_t - \bar{x})(x_t - \bar{x})'] = h_x \Sigma_x(0) h_x' + h_u \Sigma_u h_u'$$

where $\Sigma_x(0)$ is fixed point of Lyapunov equation

From this one can compute the theoretical autocovariogram $\Sigma_x(j)$ and $\Sigma_y(j)$

Higher-order approximation

Consider a univariate example, where the policy function is approximated with second-order perturbation:

$$x_t = g_x x_{t-1} + g_{xx} x_{t-1}^2 + g_u u_t, \qquad |g_x| < 1, \qquad g_{xx} > 0$$

$|g_x| < 1$ ensures that first-order approximation is stable and unique

Two fixed-points: $\bar{x} = 0$ and $\bar{x} = (1 - g_x)/g_{xx}$

$\hookrightarrow$ Once the model passes the second fixed point it explodes

Problem of higher-order approximations

Possibility of explosive behavior in higher-order approximations

$\hookrightarrow$ Model may not be stationary or does not have an ergodic probability distribution

Solution: Pruning

- Leave out terms in solution that have higher-order effects than the approximation order
- Kim, Kim, Schaumburg and Sims (2008) and Andreasen, Fernández-Villaverde and Rubio-Ramírez (2018) show that pruned state space is stationary and ergodic
- Lombardo and Uhlig (2017) or Lan and Meyer-Gohde (2013) provide theoretical foundation for this seemingly *ad-hoc* procedure

Pruning for univariate example

Decompose state vector into 1st- and 2nd-order effects

$$x_t = x_t^f + x_t^s = \; g_x x_{t-1}^f + g_x x_{t-1}^s + g_{xx}\left(x_{t-1}^f\right)^2 + 2g_{xx}\left(x_{t-1}^f x_{t-1}^s\right)^2 + g_{xx}\left(x_{t-1}^s\right)^2 + g_u u_t$$

Stable solution: Prune terms that contain $x_t^f x_t^s$ and $(x_t^s)^2$ to get law of motions:

$$x_t^f = g_x x_{t-1}^f + g_u u_t$$

$$x_t^s = g_x x_{t-1}^s + g_{xx}(x_{t-1}^f)^2$$

$$(x_t^f)^2 = g_x^2\left(x_{t-1}^f\right)^2 + 2g_x g_u x_{t-1}^f u_t + g_u^2 u_t^2$$

Univariate example

Pruned solution can be rewritten as a stable state-space system

$$\underbrace{\begin{pmatrix} x_t^f \\ x_t^s \\ x_t^{f^2} \end{pmatrix}}_{z_t} = \underbrace{\begin{pmatrix} g_x & 0 & 0 \\ 0 & g_x & g_{xx} \\ 0 & 0 & g_x^2 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} x_{t-1}^f \\ x_{t-1}^s \\ x_{t-1}^{f^2} \end{pmatrix}}_{z_{t-1}} + \underbrace{\begin{pmatrix} g_u & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 2g_x g_u & g_u^2 \end{pmatrix}}_{B} \underbrace{\begin{pmatrix} u_t \\ x_{t-1}^f u_t \\ u_t^2 \end{pmatrix}}_{\xi_t}$$

Univariate example

Pruned solution can be rewritten as a stable state-space system

$$\underbrace{\begin{pmatrix} x_t^f \\ x_t^s \\ x_t^{f^2} \end{pmatrix}}_{z_t} = \underbrace{\begin{pmatrix} g_x & 0 & 0 \\ 0 & g_x & g_{xx} \\ 0 & 0 & g_x^2 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} x_{t-1}^f \\ x_{t-1}^s \\ x_{t-1}^{f^2} \end{pmatrix}}_{z_{t-1}} + \underbrace{\begin{pmatrix} g_u & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 2g_x g_u & g_u^2 \end{pmatrix}}_{B} \underbrace{\begin{pmatrix} u_t \\ x_{t-1}^f u_t \\ u_t^2 - \sigma_u^2 \end{pmatrix}}_{\xi_t} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ g_u^2 \sigma_u^2 \end{pmatrix}}_{c}$$

Note: Even if u_t is Gaussian, ξ_t is not!

↪Higher-order statistics (HOS) may contain additional information for estimation

Pruned State Space System

Proposition (Andreasen et al, 2018):
Given an extended state vector z_t and an extended vector of innovations ξ_t, the pruned perturbation solution of a DSGE model can be rewritten as a linear time-invariant state-space system for any approximation order:

$$z_t = c + Az_{t-1} + B\xi_t$$

$$y_t = \bar{y} + d + Cz_{t-1} + D\xi_t$$

Straightforward (but tedious) to compute moments (very similar to first-order)

Method of Moments

Basic idea

In calibration, we look for parameter values that match the moments in the model to the models in the data

We can formulate this as an estimation problem: try to minimize the distance between model moments and data moments

- Simulated Method of Moments: model moments are computed via simulations

- Generalized Method of Moments: model moments are computed theoretically

Simulated Method of Moments

Simulated Method of Moments

m_t: vector of empirical observations on variables whose moments are of interest (e.g. $y_t, c_t, y_t y_{t-1}, y_t y_t y_t$)

$m_i(\theta)$: counterpart of m_t whose elements are computed on basis of artificial data generated by the DSGE model using parameter values θ

W: weighting matrix (if you have more moments than parameters)

T is sample size; τT is number of observations in artificial time series

Simulated Method of Moments

Moments distance: $G(\theta) = \frac{1}{T}\sum_{t=1}^{T} m_t - \frac{1}{\tau T}\sum_{i}^{\tau T} m_i(\theta)$

SMM estimator $\hat{\theta}_{SMM}$ is the value that minimizes $G(\theta)'WG(\theta)$

Under regularity conditions in Duffie and Singleton (1993):

$$\sqrt{T}\left(\hat{\theta}_{SMM} - \theta\right) \to N\left(0,(1 + 1/\tau)\left(D'W^{opt}D\right)^{-1}\right) \text{ with } D = E\left[\frac{\partial m_i(\theta)}{\partial\theta}\right]$$

Simulated Method of Moments

W is computed using a Newey-West estimator with a Bartlett kernel

$$D = E\left[\frac{\partial m_i(\theta)}{\partial \theta}\right]$$

- must be full rank (local identification!)
- expectation is approximated by averaging over simulated τT data points
- derivative is computed numerically

Generalized Method of Moments

Generalized Method of Moments

m_t: vector of empirical observations on variables whose moments are of interest (e.g. $y_t, c_t, y_t y_{t-1}, y_t y_t y_t$)

$E\left[m(\theta)\right]$: counterpart of m_t whose elements are computed on basis of unconditional theoretical moments of the DSGE model using parameter values θ

W: weighting matrix (if you have more moments than parameters)

T is sample size

Generalized Method of Moments

Moments distance: $G(\theta) = \frac{1}{T}\sum_{t=1}^{T} m_t - E\left[m(\theta)\right]$

GMM estimator $\hat{\theta}_{GMM}$ is the value that minimizes $G(\theta)'WG(\theta)$

Under regularity conditions in Hansen (1982):

$$\sqrt{T}\left(\hat{\theta}_{GMM} - \theta\right) \rightarrow N\left(0, \left(D'W^{opt}D\right)^{-1}\right) \text{ with } D = E\left[\frac{\partial m(\theta)}{\partial \theta}\right]$$

Generalized Method of Moments

W is computed using a Newey-West estimator with a Bartlett kernel

$$D = E\left[\frac{\partial m(\theta)}{\partial \theta}\right]$$

- must be full rank (local identification!)
- can be computed either analytically or numerically

Overidentification test

When you have more moments than parameters a general specification test of the model may be constructed (Hansen, 1982):

$$\text{SMM: } T(1 + 1/\tau)\left(G(\hat{\theta}_{SMM})' W^{opt} G(\hat{\theta}_{SMM}) \right) \rightarrow \chi^2(n_m - n_\theta)$$

$$\text{GMM: } T\left(G(\hat{\theta}_{GMM})' W^{opt} G(\hat{\theta}_{GMM}) \right) \rightarrow \chi^2(n_m - n_\theta)$$

Remarks

Comparison to full-information methods

Full-information (ML, Bayesian MCMC) methods are more efficient, but suffer more from misspecification

Limited-information methods like SMM/GMM are less efficient, but suffer less from misspecification

Efficiency of Method of Moments depends on informativeness of moments (checkout `dynare_sensitivity` and `identification`)

Penalized Method of Moments

Global minimum often found in regions of parameter space typically considered unlikely (*dilemma of absurd parameter estimates*)

Local minima often in more plausible regions and characterized by slightly worse values of the objective function

↪ include prior knowledge as an additional moment restriction (similar to adding priors to likelihood in Bayesian MCMC estimation)

$$\hat{\theta} = argmin_{\theta \in \Theta} \begin{bmatrix} G(\theta) \\ \theta^{prior} - \theta \end{bmatrix}' \begin{bmatrix} W & 0 \\ 0 & \Omega(\theta^{prior})^{-1} \end{bmatrix} \begin{bmatrix} G(\theta) \\ \theta^{prior} - \theta \end{bmatrix}$$

Caveat: leads to a loss in efficiency but may deliver good results

Dynare Implementation

Method of Moment Toolbox

New interface

- `matched_moments` block
- `method_of_moments` command

Common interface

- `varobs`
- `estimated_params`

matched_moments

$$E(c_t)$$
$$E(y_t)$$
$$E(c_t^2)$$
$$E(c_t y_t)$$
$$E(c_t c_{t-1})$$
$$E(c_t c_{t-2})$$
$$E(c_t y_{t-3})$$

```
matched_moments;
  c;
  y;
  c*c;
  c*y;
  c*c(-1);
  c*c(-2);
  c*y(-3);
end;
```

method_of_moments

`method_of_moments(NECESSARY OPTIONS)`

- `mom_method=GMM` possible values are `GMM` or `SMM`
- `datafile='MYDATA.mat'` name of filename with data

method_of_moments

`method_of_moments(OPTIONS FOR BOTH GMM AND SMM)`

- `order=2` order of perturbation approximation
  for GMM only up to 3, for SMM any order
- `penalized_estimator` include deviation from prior mean as additional moment restriction and use prior precision as weight
- `pruning` use pruning at orders>1
  automatically triggered for GMM
- `verbose` display and store intermediate estimation results

method_of_moments

`method_of_moments(OPTIONS FOR BOTH GMM AND SMM)`

- `weighting_matrix=['DIAGONAL','OPTIMAL']`
  weighting matrix in moments distance objective (does iterated estimation!)
  - `IDENTITY_MATRIX` use identitiy matrix
  - `OPTIMAL` optimal weighting matrix
  - `DIAGONAL` use diagonal of optimal weighting matrix
  - `filename` use user-provided weighting matrix

- `weighting_matrix_scaling_factor=1` scaling of weighting matrix in objective

- `se_tolx=1e-6` step size for numerical computation of std errors

method_of_moments

`method_of_moments(OPTIONS FOR SMM)`

• `burnin=500`	number of periods dropped at beginning of simulation
• `bounded_shock_support`	trim shocks in simulation to +- 2 stdev
• `seed=24051986`	seed used in simulations
• `simulation_multiple=5`	multiple of data length used for simulation

method_of_moments

`method_of_moments(OPTIONS FOR GMM)`

- `analytic_standard_errors` compute standard errors using analytical derivatives

method_of_moments

`method_of_moments(GENERAL OPTIONS)`

- `dirname='MYDIR'` directory in which to store estimation output
- `graph_format=EPS` specify the file format(s) for graphs saved to disk
- `nodisplay` do not display graphs, but save them to disk
- `nograph` do not create graphs (which implies that they are not saved to the disk nor displayed)

method_of_moments

`method_of_moments(GENERAL OPTIONS)`

- `noprint` do not print stuff to console
- `plot_priors=1` control plotting of priors
- `prior_trunc=1e-10` probability of extreme values of the prior density that are ignored when computing bounds
- `TeX` print TeX tables and graphics

method_of_moments

```
method_of_moments(DATA OPTIONS)
```

- `first_obs = 501` number of first observation
- `logdata` if data is already in logs
- `nobs = 250` number of observations
- `prefilter=0` demean each data series by its empirical mean and use centered moments
- `xls_sheet = data` name/number of sheet with data in Excel
- `xls_range = B2:D200` range of data in Excel sheet

method_of_moments

`method_of_moments(OPTIMIZATION OPTIONS)`

- `huge_number=1e7` value for replacing the infinite bounds on parameters by finite numbers. Used by some optimizers
- `mode_compute=3` specifies the optimizer for minimization of moments distance
- `additional_optimizer_steps=[13]` vector of additional mode-finders run after mode_compute
- `silent_optimizer` run minimization of moments distance silently without displaying results or saving files in between

method_of_moments

`method_of_moments(OPTIMIZATION OPTIONS)`

- `optim`: a list of NAME and VALUE pairs to set options for the optimization routines. Available options depend on mode_compute, e.g.:

- ```
  optim = ('TolFun'      , 1D-6     % termination tolerance on the function value
          ,'TolX'        , 1e-6     % termination tolerance on x
          ,'MaxIter'     , 3000     % maximum number of iterations allowed
          ,'MaxFunEvals' , 1D6      % maximum number of function evaluations
  allowed
          ,'UseParallel' , 1        % when true (and supported by optimizer) solver
                                    % estimates gradients in parallel
          ,'Jacobian'    , 'off'    % when 'off' gradient-based solvers approximate
                                    % Jacobian using finite differences;
                                    % for GMM we  can also pass the analytical
                                    % Jacobian to gradient-based solvers by setting
                                    % this 'on'
          )
  ```
  ```

method_of_moments

`method_of_moments(NUMERICAL ALGORITHMS OPTIONS)`

- `aim_solver` % Use AIM algorithm to compute perturbation approximation
- `k_order_solver` % use k_order_solver in higher order perturbation
- `dr=default` % method used to compute the decision rule; possible values
  DEFAULT
  CYCLE_REDUCTION
  LOGARITHMIC_REDUCTION
- `dr_cycle_reduction_tol=1e-7` % convergence criterion used in the cycle reduction
- `dr_logarithmic_reduction_tol=1e-12` % convergence criterion used in the logarithmic reduction
- `dr_logarithmic_reduction_maxiter=100` % maximum iterations used in logarithmic reduction

method_of_moments

`method_of_moments(NUMERICAL ALGORITHMS OPTIONS)`

- `lyapunov = DEFAULT` — % algorithm used to solve lyapunov equations; possible values `DEFAULT,FIXED_POINT,DOUBLING,SQUARE_ROOT_SOLVER`
- `lyapunov_complex_threshold = 1e-15` — % complex block threshold for the upper triangular matrix in symmetric Lyapunov equation solver
- `lyapunov_fixed_point_tol = 1e-10` — % convergence criterion used in the fixed point Lyapunov solver
- `lyapunov_doubling_tol = 1e-16` — % convergence criterion used in the doubling algorithm
- `sylvester = default` — % algorithm to solve Sylvester equation; possible values are `DEFAULT, FIXED_POINT`
- `sylvester_fixed_point_tol = 1e-12` — % convergence criterion used in the fixed point Sylvester solver

method_of_moments

`method_of_moments(NUMERICAL ALGORITHMS OPTIONS)`

- `qz_criterium=0.999999` % value used to split stable from unstable eigenvalues in reordering the Generalized Schur decomposition used for solving first order problems
- `qz_zero_threshold=1e-6` % value used to test if a generalized eigenvalue is 0/0 in the generalized Schur decomposition
- `schur_vec_tol=1e-11` % tolerance level used to find nonstationary variables in Schur decomposition of the transition matrix
- `mode_check` % plot the target function for values around the computed minimum for each estimated parameter in turn
- `mode_check_neighbourhood_size=5` % width of the window (expressed in percentage deviation) around the computed minimum to be displayed on the diagnostic plots
- `mode_check_symmetric_plots=1` % ensure that the check plots are symmetric around the minimum
- `mode_check_number_of_points=20` % number of points around the minimum where the target function is evaluated (for each parameter)

Examples

RBC_MoM_SMM_order2.mod

RBC_MoM_GMM_order2.mod