The Science and Art of DSGE Modelling - A Foundations Course

Bayesian Model Comparisons, Model Validation and Historical Decomposition

VASCO GABRIEL

September 10, 2020

The Marginal Likelihood

- Go back to Bayes Rule

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} \propto p(y|\theta)p(\theta)$$

- Suppose we have m alternative models, indexed by $i = 1, 2, ..., m$, which then depend on parameters θ^i, so we now have

$$p(\theta^i|y, m_i) = \frac{p(y|\theta^i, m_i)p(\theta^i|m_i)}{p(y|m_i)}$$

- We can express the probability of whether a model is correct or not using the usual Bayes rule:

$$p(m_i|y) = \frac{p(y|m_i)p(m_i)}{p(y)}$$

- The *posterior model probability* $p(m_i|y)$ is a function of the *prior model probability* (i.e. how likely we believe m_i to be correct before seeing the data) and the model's *marginal likelihood* $p(y|m_i)$

The Marginal Likelihood

- For a particular model i from a number of alternatives m_i we can define a density conditional on this model

$$p(y|m_i) = \int_{\Theta} p(y|\theta, m_i) p(\theta^i|m_i) d\theta$$

by integrating wrt to θ^i.

- Note that the marginal likelihood depends only on the likelihood $p(y|\theta, m_i)$ and the prior $p(\theta|m_i)$

Bayes Factor and Model Odds

- Bayesian inference now allows a framework for comparing alternative and potentially misspecified models based on their marginal likelihoods - a "likelihood race"
- Now construct a **Posterior Odds Ratio** (assuming m_i and m_j):

$$PO_{i,j} = \frac{p(m_i|y)}{p(m_j|y)} = \frac{p(y|m_i)p(m_i)}{p(y|m_j)p(m_j)}$$

- Or a **Bayes Factor** (when the prior odds ratio, $\frac{p(m_i)}{p(m_j)}$, is set to unity):

$$BF_{i,j} = \frac{p(y|m_i)}{p(y|m_j)} = \frac{\exp(LL(y|m_i))}{\exp(LL(y|m_j))}$$

defining the log-likelihood

$$LL(y|m_i) \equiv log(p(y|m_i))$$

noting that $x = \exp(\log x)$).

Bayes Factor and Model Odds - cont.

- Given the Bayes factors one can easily compute the model probabilities $p_1, p_2, ...p_n$ for n models. Since $\sum_{i=1}^{n} p_i = 1$:

$$\frac{1}{p_1} = \sum_{i=2}^{n} BF_{i,1}$$

  from which p_1 is obtained. Then $p_i = p_1 BF(i,1)$ gives the remaining model probabilities
- **model_odds.m** (or **modelcomparison.m**), computes these probabilities given the data densities from the competing models

Dynare and Matlab Files I

- **NKlinear_Est_All.mod** estimates model NKlinear with both habit and indexing. Includes Brook-Gelman convergence diagnostics, second-order stochastic simulation and historical variance decomposition.
- **NKlinear_Est_Habit.mod** estimates model NKlinear with habit, but no indexing. Use mode file from NKlinear_Est_All.mod as initial mode.
- **NKlinear_Est_Index.mod** estimates NKlinear with indexing, but no habit. Use mode file from NKlinear_Est_All.mod as initial mode.
- **NKlinear_Est_None.mod** estimates NKlinear with neither habit nor indexing (no persistence mechanisms). Use mode file from NKlinear_Est_Indexing.mod as initial mode.
- Results are given for **100,000 mcmc draws** but today you will need to restrict the number to **10,000**

Dynare and Matlab Files II

- Matlab programmes **model_odds.m** or **modelcomparison.m** compute model odd from log-likelihood values
- Matlab program **acfs_plot.m** plots the sample and estimated **Auto-Correlation Functions (acfs)**. Requires subfunctions **acfcomp.m** and **autocov.m**.
- Matlab program **irfs_plot.m** plots the impulse response functions for the estimated model.

Bayesian Model Comparison

- Formal Bayesian comparison of models **NKlinear_Est_All.mod**, **NKlinear_Est_Habit.mod**, **NKlinear_Est_Index.mod** and **NKlinear_Est_None.mod**

- Assumes a conventional Taylor rule

- Results based on LL(mcmc):

	Model All	Model Habit	Model Index	Model None
LLs(mode)	-80.14	-73.52	-93.89	-85.91
LLs(mcmc)	-72.29	-68.49	-91.03	-85.01
prob.	0.0219	0.9781	0.0000	0.0000

Table 1: Marginal Log-likelihood Values and Posterior Model Odds

- If there are mcmc convergence problems with some models it may be best to use LL(mode) as the criteria.

Limitations

- Such comparisons are important in the assessment of rival models
- A limitation is that the assessment of model fit is only relative to its other rivals with different restrictions
- The outperforming model in the space of competing models may still be poor (potentially misspecified)
- Ability of the absolute performance of one particular model against data
- Need to assess model's implied characteristics
- Model validation with data and VAR
- Forecasting performance?

Validation based on standard moment criteria

- We now compare the ability of the four estimated models to predict second moments (the absolute fit)
- To recap from Day 3: We have three sets of second moments:
  - Volatility - Standard Deviations
  - Co-Movement - Cross Correlations
  - Persistence - Autocorrelation
- To generate moments of endogenous variables in Dynare we simply use stoch_simul:
- Uses post-estimation solution based on posterior modes or means of the model to produce the three moment above.

Results and plots

- Again all simulation outputs are stored in the **FILENAME_results.mat** in the working directory ⇒ reload it to extract useful information (in the structure array *oo_*)
- e.g. the simulated autocorrelation function can be found on the diagonal of the field *oo_.autocorr*
- Need subfunctions **acfcomp.m** and **autocov.m** to compute the sample ACF.
- In the working directory, **acfs_plot_.m** plots the sample ACFs and estimated ACFs from the model
- The next Table extends the moments comparison on Day 3 (for Model All) to all four models
- We use 100,000 mcmc draws which takes time! During the day you should use 10,000 and In as
- For each criterion the best-performing model is underlined.
- **Mixed results:** each model performs best for 2/8 criteria. But for **ACFs of higher order** Slide 11 models All and Habit are clearly the best.

Validation: Matching Second Moments

	Standard Deviation		
	Output	Inflation	Interest rate
Data	0.5398	0.2400	0.6142
Model All	<u>0.5928</u>	0.3448	0.4205
Model Habit	0.5996	<u>0.3113</u>	0.4073
Model Index	0.6788	0.7212	0.6814
Model None	0.6886	0.5612	<u>0.5318</u>
	Cross-correlation with Output		
Data	1.000	-0.3199	0.0064
Model All	1.000	-0.0788	-0.2847
Model Habit	1.000	<u>-0.0969</u>	-0.2646
Model Index	1.000	0.0420	<u>-0.0245</u>
Model None	1.000	0.0673	-0.0492
	Autocorrelations (Order=1)		
Data	0.1466	0.5204	0.9371
Model All	0.0957	<u>0.5714</u>	0.8938
Model Habit	0.1150	0.5778	0.9074
Model Index	<u>0.1487</u>	0.8804	0.9667
Model None	0.1479	0.8418	<u>0.9460</u>

Autocorrelation Function Plots

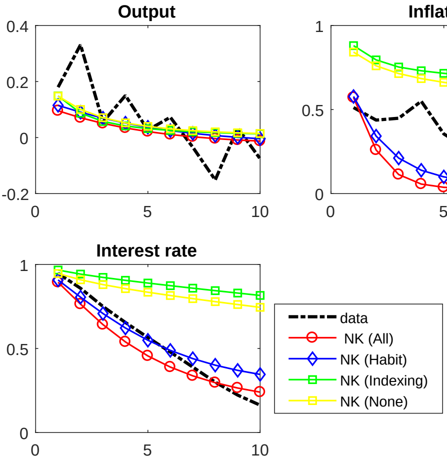

Figure 1: Autocorrelations of Observables in the Actual Data and in the Estimated Models

Impulse Response Functions (IRFs)

- Importance of shocks to the endogenous variables of interests by analysing the impulse response to the structural shocks in the models
- IRFs directly related from the state space representation $\Rightarrow$ MA representation (see later slide).
- Dynare procedure runs an IRF (that starts from the exact steady state), by sampling shocks from the distribution (with 1 s.d.), to see how the system reacts for the given periods

In Dynare:

- Simply using the *stoch_simul* keyword following estimation, and adding a list of variables of interest, e.g. *stoch_simul(irf=20) dy pinfobs robs;*, generates the IRFs using the estimated posterior means
- Again all simulation outputs from Dynare are stored in **FILENAME_results.mat**, so retrieving the field *oo_.irfs* from the above .mat file allows us to subplot and compare the IRFs from different models (see **irfs_plot.m**.)

Impulse Response Functions (IRFs)

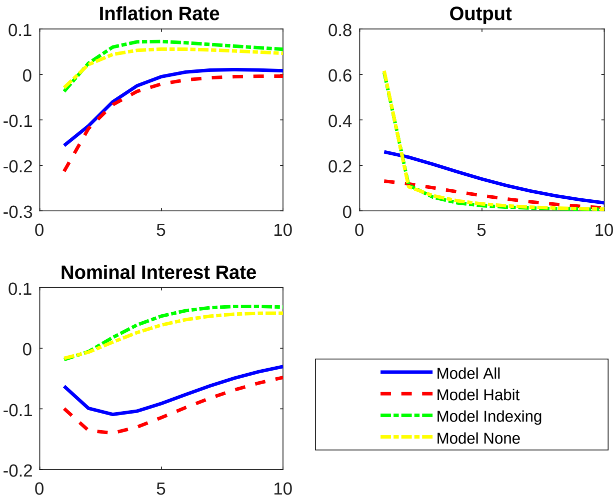

Figure 2: irfs- Technology shock

Endogenous Priors

- In many cases, the justification for the choice of priors reflects more the prior that some previous eminent researcher has got her priors right!

- Del Negro and Schorfheide (2008) proposes an easily implementable method to obtain **prior distributions** for DSGE model parameters from **data for moments of observable variables**.

- They divide the parameters into **three groups**, which reflect the information used to construct the prior:

  1. Parameters that determine **the steady states**. These are **calibrated** using the method of Day 1
  2. Parameters governing the DSGE model's endogenous propagation mechanism where prior information coming from **unrelated data set**s, e.g. the prior for the labor supply elasticity.
  3. Parameters describing the propagation mechanism of **exogenous shocks**.

Endogenous Priors For the Endogenous Shocks

- Del Negro and Schorfheide (2008) propose a method of "endogenous priors" that translates priors for second moments of observables into a joint prior distribution for these parameters.
- Such priors may come from pre-sample evidence, for instance, and are assumed to be invariant across different DSGE model specifications.
- Dynare implements a version of this procedure due to Christiano *et al.* (2011) except the "pre-sample" is the actual sample.
- Simply add **endogenous_prior** to the estimation command.
- The product of the initial priors and the pre-sample likelihood of the standard deviations of the observables is used as the new prior.
- This is really a **hydrid** Simulated Method of Moments (SMM) - Bayesian Estimation Procedure.
- See the technical appendix of Christiano *et al.* (2011).
- The following table indicates the endogenous prior feature produces *some* improvement in moment matching.

Dynare

- **NKlinear_Est_All_End_Priors.mod** estimates model NKlinear with both habit and indexing. Use mode file from NKlinear_Est_All.mod as initial mode.
- **NKlinear_Est_Habit_End_Priors.mod** estimates model NKlinear with habit, but no indexing. Use mode file from NKlinear_Est_Habit.mod as initial mode.
- **NKlinear_Est_Indexing_End_Priors.mod** estimates NKlinear with indexing, but no habit. Use mode file from NKlinear_Est_Indexing.mod as initial mode.
- **NKlinear_Est_None_End_Priors.mod** estimates NKlinear with neither habit nor indexing (no persistence mechanisms). Use mode file from NKlinear_Est_None.mod as initial mode.
- Results are given for **100,000 mcmc draws** but today you will need to restrict the number to **10,000**

Matching Second Moments with Endogenous Priors

Standard Deviation			
	Output	Inflation	Interest rate
Data	0.5398	0.2400	0.6142
Model All (Previous)	0.5928	0.3448	<u>0.4205</u>
Model All (End. Priors)	<u>0.5776</u>	<u>0.2738</u>	0.3847
Cross-correlation with Output			
Data	1.000	-0.3199	-0.0064
Model All (Previous)	1.000	-0.0788	-0.2847
Model All (End. Priors)	1.000	<u>-0.0692</u>	-0.2148
Autocorrelations (Order=1)			
Data	0.1466	0.5204	0.9371
Model All (Previous)	<u>0.0957</u>	0.5714	0.8938
Model All (End. Priors)	0.0603	<u>0.5432</u>	<u>0.8890</u>

Variance and Historical Decompositions

- Variance decompositions decomposes the variation in each endogenous variable into each shock to the system, thus providing information on the the relative importance of each disturbance as a source of variation for each variable
- Historical decompositions can be used for counterfactual simulations.
- The data can be decomposed into the sum of a baseline forecast and the contribution of all shocks. This allow us to analyse how the data would have evolved if a shock or a combination of shocks are shut down (i.e., their contribution is zero)

State Space Representation Again

- Recall the state-space representation of the model solution

$$X_{t+1} = AX_t + B\varepsilon_{t+1} \quad (1)$$
$$Y_t = CX_t \quad (2)$$

where X_t is the potentially unobservable state (column) vector, Y_t is the vector of the observables (data) and ϵ_t is a vector of shocks.

- Solving (1) backwards, recursively we have

$$Y_t = CX_t = C(AX_{t-1} + B\varepsilon_t) = C(A(AX_{t-2} + B\varepsilon_{t-1}) + B\varepsilon_t)$$

- Hence continuing with this recursive process we arrive at

$$Y_t = \sum_{j=0}^{t} CA^j \epsilon_{t-j} + CA^{t+1} X_0 \quad (3)$$

- The historical decomposition stems from this **Moving Average (MA)** representation of the model state space

Moving Average Representation of the Solution

- To recap: the MA representation of the model state space is

$$Y_t = \sum_{j=0}^{t} D_j \varepsilon_{t-j} + CA^{t+1} X_0$$

  for the data sample $t = 1, ..., T$. For each matrix D_j, denote its ith row multiplying the ith shock by $d_{i,j}$.

- If we further define the effect of the ith shock on Y_t as $Y_{i,t} = \sum_{j=0}^{t} d_{i,j} \varepsilon_{i,t-j}$, then we can decompose Y_t as

$$Y_t = \sum_{i=0}^{r} Y_{i,t} + CA^{t+1} X_0 \tag{4}$$

  where r is the number of shocks.

- This is what historical decomposition in Dynare produces.

Variance and Historical Decompositions in Dynare

- Dynare calculates each of the individual terms of (4), with the last term, $CA^h X_0$, shown on the historical decomposition graphs as the effect of 'initial values'.
- The decomposition for a given sample according to the model can be computed using the command *shock_decomposition* which must be followed by the *estimation* statement: *shock_decomposition (parameter_set=posterior_mode) dy pinfobs robs;*
- Example: using the NK linear model estimated in the earlier section - **NKlinear_Est_HistDecom.mod**

Historical Decomposition of Inflation

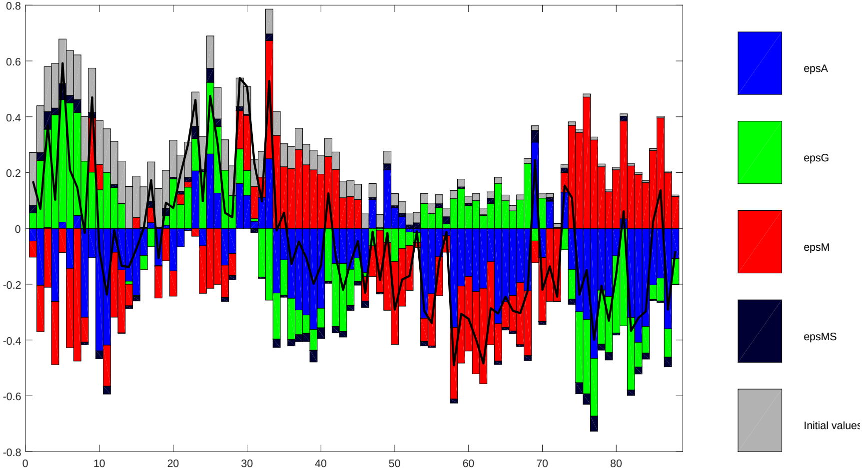

Figure 3: Historical Decomposition of Inflation: Model All

Estimation, Comparison and Validation; Summary

- The **choice of filter** to make the data stationary - see Course Notes on removing trends. We use differencing for output.
- The **measurement equation**: this links the data with the output of the model
- the choice of **priors**: depends on the range of possible values for the parameter. General guidance: *inverse gamma* distributions for non-negativity constraints, *beta* distributions for fractions or probabilities, *normal* distributions when more informative priors are necessary (*uniform* or 'flat' priors if there is little information about the parameter)
- Computation of **Posterior**: Bayes theorem, mode computation and MCMC.
- **Model comparison**: construct the Bayes Odds as above.
- **Model validation**: compare second moments with those of the data.
- **Endogenous Priors** give *some* improvement in matching second moments.
- **DSGE-VAR & Forecasting**: further model comparison and empirical validation (tomorrow)

Exercises on Dynare I: Estimation and Comparison

1. Repeat the model comparison exercise across for model specifications (all persistence mechanisms, only habit, only indexing and no persistence mechanisms) using an **implementable rule** rather than the conventional Taylor rule.
2. For today's exercise find the posterior distribution based on 10,000 of sets MCMC-MH simulation. But after the course repeat the process with 100,000 draws.
3. Use mode files from the conventional Taylor rule as initial modes in the estimation command. In most cases mode_compute=4 should give a negative definite Hessian. If not you must use mode_compute=6 which is designed to compute a negative definite Hessian, but it takes a lot of time!
4. When you get this far you have LLs for eight models. Now conduct a likelihood race across these eight models.
5. Since you use only 10,000 draws there will be mcmc convergence problems with some models, so use LL(mode) as the criteria.

Exercises on Dynare II: Validation

1. Now repeat the moment comparison exercise using an **implementable rule** rather than the conventional Taylor rule.
2. From the stochastic simulation of the estimated models with an implementable monetary rule obtain the model-implied moments based on the estimated posterior modes and produce the IRFs.
3. Use these results validate all eight model variants against the real word data by extending the Table on Slide 10
4. Use **acfs_plot.m** to compare autocorrelations of the observables in the actual data and in the eight estimated models. (These use MATLAB files **acfcomp.m**, **autocov.m** in the folder)
5. Use **irfs_plot.m** to compare the impulse response functions of the eight estimated models.

Christiano, L. J., Trabandt, M., and Walentin, K. (2011). Introducing financial frictions and unemployment into a small open economy model. *Journal of Economic Dynamics and Control*, **35**(12), 1999—-2041.

Del Negro, M. and Schorfheide, F. (2008). Forming Priors for DSGE models (and how it affects the Assessment of Nominal Rigidities. *Journal of Monetary Economics*, **55**(7), 1191–1208.