Readme for Estobin

Matteo Iacoviello
Federal Reserve Board

June 3, 2018

Notes for Estimation. This file describes how to implement the estimation in Occbin.
Note: the estimation uses the toolkit 20130531 and Dynare 4.3.1

1. Folders that need to be added to Matlab path

1. `Dynare 4.3.1\matlab` (included in this distribution)
2. `occbin_20130531\toolkit_files` (included in this distribution)
3. `estobin`

2. Notes about Estimation in Occbin: Key files

1. `posterior1.m/posterio2.m`

   Computes the likelihood or posterior of a model given parameter values, observables, shocks, choice of priors.

   function [posterior filtered_ errs like prior resids] =posterior1(current_ params,params_ labels,

   lowerbound, upperbound,...

   modnam_ 00_ ,modnam_ 10_ ,modnam_ 01_ ,modnam_ 11_ ,...

   constraint1_ difference, constraint2_ difference,

   constraint_ relax1_ difference, constraint_ relax2_ difference,...

   err_ list,obs_ list,obs,ntrain,codes, p6, p7, IPRIOR,solver)

   solver=3 uses the Francesco Ferrante Filter for models with one occasionally binding constraint only.

2. `myfilter.m`

   Filters the shocks that across the entire sample from 1 to T are consistent with given realizations of the observables

   function [filtered_ errs resids Emat] = myfilter(constraint1_ difference, constraint2_ difference,...

   constraint_ relax1_ difference, constraint_ relax2_ difference,err_ list,obs_ list,obs,solver);

   option solver=3 above uses `franco_filter_posterior` to filter the shocks

3. `match_function.m`

   In the vector resids, the function returns in the vector `resids` the distance between observables and data in period current_obs for a given value of shocks and initial conditions.

   function [resids grad init_ out E newviolvecbool relaxconstraint iter] = match_function(...

   err_ vals,err_ list,obs_ list,current_ obs,init_ val,...

   constraint1,constraint2,constraint_ relax1,constraint_ relax2)

3. Other things to keep in mind

1. `call_pre_estimation_script.m`

   % This is a script that prior to estimation does the following

   % 1) Select variables used in estimation

   % 2) creates eval_zdata_script.m file

   % 3) creates eval_endo_names.m file containing steady-state values of variables

   % 4) creates eval_param.m containing parameter values

   % 5) processes constraint and creates eval_difference_script

4. Example Models

Two examples.

Model with Borrowing Constraint as in Appendix C of JME 2015, https://www2.bc.edu/matteo-iacoviello/research_files/JME_TA_2015.pdf

Model with ZLB which is essentially the canonical three-equation NK model.

4.1. Model with Borrowing Constraint and Francesco Ferrante's Filter: borrcon_francofilter_distr

1. Generate artificial data using `run_generate_artificialdata`
2. Artificial data and shocks are saved in mat file `artificial_data` in the same folder
3. For occbin estimation – mode –, type "`run_likelihood_artificialdata.m`", using option `method = 'fminsearch'`
4. After finding the mode, switch to `method = 'metropolis'` to run metropolis-hastings algorithm and to estimate distribution of parameters.
5. Parameters are set as follows

   R = 1.05;BETA = 0.945;RHO = 0.9;STD_U = 0.01;M = 1;GAMMAC = 3;
6. Estimation results after 1,000 draws are as follows (using IPRIOR=1 which does Bayesian estimation.)

```
C:\Dropbox\E\occbin_Estimation_web\borrcon_francofilter_distr
Number of runs is 1000

PARAMETER   MODE MODE-METRO    PRIOR-DIS    PRIOR-MEAN PRIOR-SD POST-MEAN  10%      50%     90%
GAMMAC     2.6074   2.6074      NORMAL_PDF     3.0000    1.0000      2.6939   2.0040   2.7198   3.4122
STD_U      0.0062   0.0062      INV_GAMMA_PDF  0.0100    1.0000      0.0063   0.0057   0.0063   0.0071

Calibrated parameters
------------------
BETA  =     0.945000;
M     =     1.000000;
R     =     1.050000;
RHO   =     0.900000;


Laplace approximation
Marginal data density is 400.5267
```

4.2. Model with Borrowing constraint: folders borrcon, borrcon_alt1, borrocon_alt

1. Generate artificial data using `runsim_borrcon.`
2. Artificial data and shocks are saved in mat file `fakedata` in the same folder
3. For occbin estimation, type "`run_likelihood.m`". Make sure you line up calibrated parameters across `runsim_borrcon` and `run_likelihood.m`
4. After estimation, file `call_check_metropolis` generates nice table with results.

   Parameters set as follows

   $R = 1.05$, $\beta = 0.945$, $\rho = 0$, $m = 1$

	Calib	Estim (10%,50%,90%)	Calib	Estim (10%,50%,90%)	Calib	Estim (10%,50%,90%)
σ_u	0.01	0.0084, 0.0097, 0.0111	0.01	0.0086, 0.0096, 0.0109	0.01	0.0095, 0.0107, 0.0120
γ	**3**	2.0165, **2.7933**, 3.3085	**2**	1.5564, **1.7618**, 2.0949	**5**	4.9537, **5.6825**, 6.0206

4.3. Model with ZLB: folders cgg_rlong and cgg_rpol

1. Mod files: `cgg.mod and cgg_zlb.mod`. Calibrated parameters are set in `paramfile_cgg`
2. To see how model works, type `runsim_cgg.m`
3. Generate artificial data using `runsim_cgg_generate_fakedata.`
4. Artificial data and shocks are saved in mat file `fakedata` in the same folder
5. For occbin estimation, run "`run_likelihood_cgg.m`".
6. Notes: the estimation uses the long rate as an observable to avoid the invertibility problem that would prohibit solving for unique shocks when the economy is at the ZLB.

   The estimation using r as observable follows the methods in Guerrieri–Iacoviello JME, and tweaks the model by assuming that there are no monetary shocks at the ZLB!

Parameters are set as follows:
LAMBDA=0.04;
BETA=0.9925;
FIP=4;
FIY=4;
FIR=0.8;
PHI=1;
RHOY=0.5;
STD_P=0.001;
STD_R=0.001;
STD_Y=0.02;
To avoid invertibility of the likelihood problems, we use rlong as an observable in estimation.

		Using rlong as observable	Using rpol as observable and the methods in GI, JME 2017
	Calib	Estim (10%,50%,90%)	
ρ_y	0.5	0.4987, 0.4995, 0.5003	TBA
σ_p	0.001	0.0011, 0.0012, 0.0013	
σ_R	0.001	0.0009, 0.0010, 0.0011	
σ_y	0.02	0.0189, 0.0208, 0.0228	