Alastair Hall April 2016

GMMGUI: Introduction

These notes explain how to the MATLAB toolbox for GMM estimation written by Kostas Kyriakoulis. This toolbox has a *Graphical User Interface* (GUI) that greatly simplifies implementation. We introduce the main features of the GUI using Hansen and Singleton's (1982) consumption based asset pricing model; see Hall (2005)[Chapter 1.3.1] for a description of the model. Here, we consider a version of the model in which the population moment condition takes the form:

$$E[f(v_t, \theta)] \;=\; E[e_t(\gamma, \delta)\, z_t] \;=\; 0,$$

where

$$e_t(\gamma, \delta) \;=\; \delta x_{1,t}^{\gamma-1} x_{2,t} - 1,$$

$x_{1,t} = c_{t+1}/c_t$, $x_{2,t} = r_{t+1}/p_t$, $z_t = (1, x_{1,t-1}, x_{1,t-2}, x_{2,t-1}, x_{2,t-2})'$, c_t is aggregate per capita consumption, and r_{t+1}, p_t are respectively the gain in period $t+1$ and the price in period t on a stock index. The data is monthly and the sample period is 1960.1–1997.12. This version of the model is the running example in Hall (2005).

First, download the zip file containing the GMMGUI files, and extract all the files to a directory (that must appear in your MATLAB *path* for the files to be accessed). Before opening the GMMGUI, it is convenient to first load the data for this example. The data files are available in the zip file **GMMGUI_cbapm**; download these to a directory in your MATLAB *path.* Below, we work with the value weighted NYSE index (VWR), but data for the equally weighted NYSE index (EWR) are also provided in the zip file. In the MATLAB Command Window enter the following:[1]

x = load('cbapmvwrdata.dat');
z = load('cbapmvwrinstr.dat');

Now enter in the MATLAB command window, the following: **gmmgui**. This causes a window to open. We now go through the these boxes in this window in turn.

- *Dataset*: Enter the data used in the model. For our example, this is **x**.
- *Moments*: Enter the name of a function m-file that calculates the moments and the Jacobian. For our example, this is **cbapm**. If you open **cbapm.m** in the MATLAB editor, then it can be seen that this file outputs two arguments: **cbapmmom**, a $T \times q$ matrix whose t^{th} row is $f(v_t, \theta)'$; **cbapmgrad**, the $q \times p$ matrix $T^{-1} \sum_{t=1}^{T} \partial f(v_t, \theta)/\partial\theta'$.
- *Various arguments*: This box is to allow entry of any other data that might be used to, for instance, calculate the weighting matrix. For models in which the moment condition involves

[1]To execute, the commands in the handout with EWR just change "v" to "e" in the following load commands.

instruments - such as in our example - then the data matrix for the instruments must be entered here. So enter **z**.

- *Starting values*: This must be a $p \times 1$ vector of starting values for the parameters. Enter **5*ones(2,1)**.
- *1st step W*: here you enter a MATLAB expression for the first step weighting matrix. Enter $\mathbf{inv}(\mathbf{z}' * \mathbf{z}/\mathbf{465})$. (In this example, $T = 465$.)
- *Cov. matrix settings*: Click on *Select Covariance matrix* and you can see three option. The *martingale difference* choice equates to the Eicker-White estimator. Choose that option.
- *GMM iterations*: This box controls aspects of the GMM estimation. *GMM iterations* sets the ceiling for the number of steps in iterated GMM estimation; *Tolerance for GMM iterations* sets the convergence criterion for stopping these iterations that is, ϵ where the iterations stop if $\|\hat{\theta}(i) - \hat{\theta}(i-1)\| < \epsilon$ where $\hat{\theta}(i)$ is the GMM estimator after the i^{th} iteration. If you right click in the boxes the default values are used.[2]
- *MATLAB's minimization procedure*: This box sets options that control the optimization procedure used to obtain $\hat{\theta}(i)$ on each i. If you right click on these boxes then the default values are used.
- *Display*, *Diagnostics* and *Derivative check*: control various issues pertaining to the estimation. Further details are in the GMMGUI help file.
- *Output will be saved as*: Output is saved in a "structure object" with this name of which more below. Enter **cbapmvwr**.

To run the estimation, click on *Estimate*. When the estimation is complete a window opens containing the parameter estimators, approximate 95% confidence intervals and the overidentifying restrictions test (J-test) along with its p-value , *c.f.* Hall (2005) Table 3.7 (p.92), Table 5.1 (p.153).

A number of other statistics associated with the estimation are stored using the "structure object" (mentioned above) so they can be used in MATLAB calculations. To illustrate, enter the following in the MATLAB commands window: **cbapmvwr.Estimates**, (hit enter), **cbapmvwr.Jtest**.

The complete list of available outputs are as follows with the term in italics providing the word after the period:

[2]Note that a warning message is printed if this criterion is not met. So for example, if we set *GMM iterations* to be 2 for the two-step estimator and use the default value for *Tolerance for GMM iterations* then the warning message will appear if the distance between the first- and second- step GMM estimators is bigger than the tolerance. This should not be confused with non-convergence of the numerical optimization on the second-step GMM estimation.

- *Estimates*: the final GMM estimates (displayed as a row vector).
- *Jtest*: the overidentifying restrictions test.
- *MomentsVariance*: the long-run variance of the moments, evaluated at the final estimates.
- *ProbJ*: the probability value of the J-test.
- *Final_Moments*: the moments' vector, evaluated at the final estimates.
- *Final_Moments_gradient*: the gradient of the moments, evaluated at the final estimates.
- *ThetaVariance*: the variance of the final GMM estimates.
- *ThetaStd*: the standard deviation of the final GMM estimates.
- *ConfInterval*: 95% confidence intervals of the estimates.
- *Dataset*: the dataset used for estimation.
- *Var_arg*: additional arguments used for estimation.
- *moments*: the name of the moments' function.
- *OPTION_StartVals*: the starting values used for the estimation.
- *OPTION_FirstStepW*: the first step weighting matrix used for the estimation.
- *OPTION_MomCovType*: the type of the long-run covariance matrix (Serially Uncorrelated, etc.).
- *OPTION_Bandwidth*: the value of bandwidth used.
- *OPTION_MaxGMMiter*: the upper bound of the GMM iterations.
- *OPTION_GMM_ToleCrit*: the tolerance criterion for the GMM iterations.
- *OPTION_MaxFunEvals*: the upper bound on function evaluations (used by Matlab's min.procedure).

To illustrate, calculate the overidentifying restrictions test directly from the estimated sample moment and the variance of the sample moment fucntion and then compare to the value reported in the output window.

The GMMGUI can also be used to implement Continuous Updating GMM estimation. This is achieved by choosing the *Estimation* option to be *Continuous Updated*. All else is the same except note now that the *1st weighting matrix* now fades to grey as it is not relevant to the estimation. Estimates for the model with VWR using Continuous updating GMM can be compared to Hall (2005) Table 3.9 (p.106).

We now explore how to implement certain other hypothesis tests using the *GMMGUI*. We first consider tests of nonlinear restrictions on the parameters of the form $r(\theta_0) = 0$ and then structural stability tests.

Testing nonlinear restrictions on the parameters:

We can test hypotheses of the type $H_0 : r(\theta) = 0$ using Wald, LM or D tests; see Hall (2005)[Ch 5.3] for a description of the statistics and discussion of their statistical properties. The latter two involve restricted GMM estimation. So we first discuss how to implement restricted GMM in the *GMMGUI*. To illustrate, we consider the restriction:

$$r(\theta) \;=\; \gamma \;=\; 0,$$

under which the CRRA utility function becomes the log utility function.

On the main *GMMGUI* page in the top left is a box marked *Estimation*. By default, it is set to *Unrestricted*. We now change this to *Restricted*. When we do so, another box opens in the lower left hand side of the main page. Therein, the top row of boxes provides a lower bound (**LB**) and upper bound (**UB**) that can used to restrict the parameter space, $LB < \theta < UB$. (This feature is not needed in our example and in such cases enter **N/A** in both the *LB* and *UB* boxes.) The second row contains a single box into which must be put the name of the m-file that calculates both $r(\theta)$ and $\partial r(\theta)/\partial\theta'$.

Enter **N/A** in both LB and UB, and insert the name **cbapmtest** into this box. Then complete the remaining boxes as for the unrestricted estimation; specify the output to be saved in **cbapmvwr_res**. Then run the estimation.

To formally test whether the restrictions hold, we access the *Testing* box and choose the option *W, LM, D*. Another window opens in which we: select the tests required; enter the output variables containing the *Unrestricted* and *Restricted* results (for our example, **cbapmvwr** and **cbapmvwr_res** respectively); enter the *constraints function*, here **cbapmtest**; enter the *number of restrictions*, 1 here; and the name of an *Output* variable, make that **logutest**. Then hit *Calculate tests*.

The output **logutest** has three fields: *Waldtest*, *LMtest* and *Dtest*. Each has two elements: the value of the test statistic and its p-value. The results can be compared to Hall (2005) Table 5.3 (p.165)

Structural stability tests: In overidentified models, the hypothesis of structural stability can be decomposed into two parts: (i) the stability of the identifying restrictions (or equivalently the stability of the parameters); (ii) the stability of the overidentifying restrictions. Part (i) can be tested using tests based on the Wald, LM and D principles; part (ii) can be tested using so-called O, O1 and O2 tests based on the sub-sample overidentifying restrictions; see Hall (2005)[Ch 5.4] for a discussion of these tests and their large sample properties. To illustrate how to calculate these statistics in the GMMGUI, we first re-estimate the model for VWR as per the instructions above. For simplicity, just do a two-step estimation. Save the output in **cbapmvwr**.

Now, in *Testing*, choose the option *Stability*. Another window opens: this has two versions depending on whether the option *Known* or *Unknown* break-point is chosen. Consider first *Known*: in this case the user must supply the break-point. We test whether there is a break associated with the change in the Federal Reserve's operating procedures in October 1979; so the break-point is observation **248**. Save the output as **cbapmvwr_fb**. Calculate the tests.

The fields for the output variable are as follows:

- *Estimates_Before* : a $p \times 1$ vector with the estimated value of the parameters before the break.
- *Estimates_After* : a $p \times 1$ vector with the estimated value of the parameters after the break.
- *Wtest*: the value and p-value of the Wald structural stability test.
- *LMtest*: the value and p-value of the LM structural stability test.
- *Dtest*: the value and p-value of the D structural stability test.
- *Otest*: the value and p-value of the O structural stability test.
- *O1test*: the value and p-value of the O1 structural stability test.
- *O2test*: the value and p-value of the O2 structural stability test.

The results can be compared to Hall (2005) Table 5.4 (p.178).

Now choose the option *Unknown*: in this case, the user must supply the lower and upper bound on the interval over which the break is allowed to occur. This bound is specified in terms of the fraction of the sample associated with the bound in question. Enter **0.15** and **0.85**. Save the output as **cbapmvwr_ub**. Calculate the tests.

The fields for the output variable are as follows:

- *Estimates*: the first p rows on each column are the estimates before the break, and the last p are the estimates after the break.
- *Wtests*: the sup-, Av-, and Exp- value of the sequence of Wald tests.
- *LMtests*: the sup-, Av-, and Exp- value of the sequence of the LM tests.
- *Dtests*: the sup-, Av-, and Exp- value of the sequence of the D tests.
- *Otests*: the sup-, Av-, and Exp- value of the sequence of the O tests.

The limiting distributions of the unknown break-point tests are non-standard and percentiles are not provided in the GMMGUI. They are available for certain choices of p (the number of parameters) and $q - p$ (the number of overidentifying restrictions) in Hall (2005) Tables 5.5 (p. 181) and 5.6 (p.183) respectively. More complete tables are available in the journal articles in which the tests are proposed. The empirical results can be compared to those reported in Hall (2005) Table 5.7 (p.186).

References

Hall, A. R. (2005). *Generalized Method of Moments*. Oxford University Press, Oxford, U.K.

Hansen, L. P., and Singleton, K. S. (1982). 'Generalized instrumental variables estimation of non-linear rational expectations models', *Econometrica*, 50: 1269–1286.