SMM: A Simple Example©

Dean Corbae

October 25, 2021

A Simple Example

- To illustrate the theory from the previous slides, here is an example based on Michaelides and Ng (2000, Journal of Econometrics).
- Take the true data generation process to be a $k = 1$ MA(1) process (recall k is the dimension of observable variables in the data)

  $$x_t = \varepsilon_t - b_0 \varepsilon_{t-1}, \varepsilon_t \overset{i.i.d}{\sim} N(0,1) \tag{1}$$

  with $\ell = 1$ parameter $b_0 = 0.5$ and $\varepsilon_0 = 0$.
- We will take the model generation process to be

  $$y_t(b) = e_t - b e_{t-1}, \quad e_t \overset{i.i.d}{\sim} N(0,1) \tag{2}$$

  with $\ell = 1$ parameter b (recall ℓ is the dimension of the parameter vector) and $e_0 = 0$.
- We do not know the true parameter value b_0 so will estimate it via simulated method of moments.

Moments

- Let m denote the mapping from some $k \times 1$ vector z_t (which could be true data (x_t) or simulated data ($y_t(b)$)) to an $n \times 1$ moment vector.
- Just for example, suppose we take $k = 1$ and consider $n = 4$ moments: mean, variance, first order autocorrelation, and second order autocorrelation given by:

$$m(z_t) = \begin{bmatrix} z_t \\ (z_t - \bar{z})^2 \\ (z_t - \bar{z})(z_{t-1} - \bar{z}) \\ (z_t - \bar{z})(z_{t-2} - \bar{z}) \end{bmatrix}. \tag{3}$$

- Thus with $n > k$ we have an overidentified model (remember I'm just trying to illustrate things).

Population Moments

- Note that we can write the population (unconditional) moment vector for the true data using m as $\mu(x) = E[m(x)]$.
- In our simple $n = 4$ example, the population data moments are:

$$\mu(x) = \left[\begin{array}{c} E\left[\varepsilon_t\right] - b_0 E\left[\varepsilon_{t-1}\right] \\ E\left[\left(\varepsilon_t - b_0\varepsilon_{t-1}\right)^2\right] \\ E\left[\left(\varepsilon_t - b_0\varepsilon_{t-1}\right)\left(\varepsilon_{t-1} - b_0\varepsilon_{t-2}\right)\right] \\ E\left[\left(\varepsilon_t - b_0\varepsilon_{t-1}\right)\left(\varepsilon_{t-2} - b_0\varepsilon_{t-3}\right)\right] \end{array} \right]$$

$$= \left[\begin{array}{c} E\left[\varepsilon_t\right] - b_0 E\left[\varepsilon_{t-1}\right] \\ E\left[\varepsilon_t^2\right] - 2b_0 E\left[\varepsilon_t\varepsilon_{t-1}\right] + b_0^2 E\left[\varepsilon_{t-1}^2\right] \\ E\left[\varepsilon_t\varepsilon_{t-1}\right] - b_0 E\left[\varepsilon_t\varepsilon_{t-2}\right] - b_0 E\left[\varepsilon_{t-1}^2\right] + b_0^2 E\left[\varepsilon_{t-1}\varepsilon_{t-2}\right] \\ E\left[\varepsilon_t\varepsilon_{t-2}\right] - b_0 E\left[\varepsilon_t\varepsilon_{t-3}\right] - b_0 E\left[\varepsilon_{t-1}\varepsilon_{t-2}\right] + b_0^2 E\left[\varepsilon_{t-1}\varepsilon_{t-3}\right] \end{array} \right] \tag{4}$$

Population Moments -cont.

- Given the i.i.d. assumption, the population data moments are then:

$$\mu(x) = \begin{bmatrix} 0 \\ 1+b_0^2 \\ -b_0 \\ 0 \end{bmatrix}. \tag{5}$$

and the population model moments

$$\mu(y(b)) = \begin{bmatrix} 0 \\ 1+b^2 \\ -b \\ 0 \end{bmatrix}. \tag{6}$$

Moment conditions

- The n moment conditions we are going to use in SMM is given by

$$u(x_t, b) = m(x_t) - \frac{1}{H}\sum_{h=1}^{H} m(y_t^h(b)) \tag{7}$$

where H is the number of simulations of the model.

- Then define an $nq \times 1$ vector (where $n = 4$ and $q = 1$) as in our previous notes

$$g(b) \equiv E[u(x_t, b)] = E\left[m(x_t)\right] - \frac{1}{H}\sum_{h=1}^{H} E\left[m(y_t^h(b))\right]$$

$$= \mu(x) - \mu(y(b))$$

Global Identification

- Note that **Global Identification** requires $g(b) = 0 \iff b = b_0$.
- In this example, the Global Identification condition is that there is a unique solution $b = b_0$ to the following equation

$$g(b) = \mu(x) - \mu(y(b)) = 0$$

$$\iff \mu(x) = \begin{bmatrix} 0 \\ 1 + b_0^2 \\ -b_0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 + b^2 \\ -b \\ 0 \end{bmatrix} = \mu(y(b)). \qquad (8)$$

- In particular, in (8) the mean and second autorcorrelation do not identify b, the variance does not uniquely identify it since $b = \pm b_0$, while the autocorrelation uniquely identifies $b = b_0$.

Optimal Weighting Matrix

- We next obtain an analytic expression for the variance-covariance matrix S and hence the optimal weighting matrix $W^* = S^{-1}$.
- Let S denote the $n \times n$ asymptotic var-covar matrix of the n moment conditions u at the true parameter value $b = b_0$:

$$S = \sum_{j=-\infty}^{\infty} E[u(x_t, b_0)u(x_{t-j}, b_0)'] \tag{9}$$

- From (7), we know

$$S = \sum_{j=-\infty}^{\infty} E\left\{\left[m(x_t) - \frac{1}{H}\sum_{h=1}^{H} m(y_t^h(b_0))\right]\left[m(x_{t-j}) - \frac{1}{H}\sum_{h=1}^{H} m(y_{t-j}^h(b_0))\right]'\right\} \tag{10}$$

Optimal Weighting Matrix - cont.

Under the assumption the 2nd moment of the true data and model are equal at the parameter $b = b_0$, then

- x_t and $y_t^h(b_0)$ have the same population mean (i.e.$\mu(x) = \mu(y(b_0))$) and
- the same var-covar matrix:

$$\begin{aligned} S_x &\equiv \sum_{j=-\infty}^{\infty} E[\{m(x_t) - \mu(x)\}\{m(x_{t-j}) - \mu(x)\}'] \\ &= \sum_{j=-\infty}^{\infty} E[\{m(y_t^h(b_0)) - \mu(y(b_0))\}\{m(y_{t-j}^h(b_0)) - \mu(y(b_0))\}'] \equiv S_y \end{aligned}$$

where S_x (S_y) is the asymptotic var-covar matrix of $m(x_t)$ (or $m(y_t^h(b_0))$).

Optimal Weighting Matrix - cont.

- In that case, adding $\mu(x)$ and subtracting $\mu(y(b_0))$ in (10) one can show (see Appendix of notes):

$$S = \left(1 + \frac{1}{H}\right) S_x .$$

given that data draws in x_t and simulation draws in $y_t^h(b_0)$ are uncorrelated and there is no correlation between $\{y_t^h(b_0)\}_{h=1}^H$ across H.

- Let $\Gamma_j \equiv E[\{m(x_t) - \mu(x)]\}\{m(x_{t-j}) - \mu(x)]\}']$ denote the j-th autocovariance of m. Then the asymptotic variance-covariance matrix of $m(x_t)$ is given by

$$S_x = \Gamma_0 + \sum_{j=1}^{\infty}(\Gamma_j + \Gamma_j') \tag{11}$$

Optimal Weighting Matrix - cont.

- Because we know the true DGP, we can compute Γ_js analytically:

$$\Gamma_0 = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & 2\sigma_x^4 & -2b_0\sigma_x^2 & 0 \\ 0 & -2b_0\sigma_x^2 & \sigma_x^4 + b_0^2 & -b_0\sigma_x^2 \\ 0 & 0 & -b_0\sigma_x^2 & \sigma_x^4 \end{bmatrix}$$

$$\Gamma_1 = \begin{bmatrix} -b_0 & 0 & 0 & 0 \\ 0 & 2b_0^2 & 0 & 0 \\ 0 & -2b_0\sigma_x^2 & b_0^2 & 0 \\ 0 & 2b_0^2 & -b_0\sigma_x^2 & b_0^2 \end{bmatrix}$$

with Γ_j zero $\forall j \geq 2$ and where $\sigma_x^2 = 1 + b_0^2$ is the variance of x_t.

Optimal Weighting Matrix - cont.

- Then the asymptotic var-covar matrix is:

$$
\begin{aligned}
S_x &= \Gamma_0 + \Gamma_1 + \Gamma_1' \\
&= \begin{bmatrix} (1-b_0)^2 & 0 & 0 & 0 \\ 0 & 2(1+4b_0^2+b_0^4) & -4b_0(1+b_0^2) & 2b_0^2 \\ 0 & -4b_0(1+b_0^2) & 1+5b_0^2+b_0^4 & -2b_0(1+b_0^2) \\ 0 & 2b_0^2 & -2b_0(1+b_0^2) & 1+4b_0^2+b_0^4 \end{bmatrix}
\end{aligned}
$$

- Evaluated at $b_0 = 0.5$,

$$
S_x = \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 4.125 & -2.5 & 0.5 \\ 0 & -2.5 & 2.3125 & -1.25 \\ 0 & 0.5 & -1.25 & 2.0625 \end{bmatrix}
$$

Optimal Weighting Matrix - cont.

- Voila! The inverse of the numerical var-covar matrix S is the optimal weighting matrix for our hypothetical MA(1):

$$W^* = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 1.1115 & 1.5705 & 0.6823 \\ 0 & 1.5705 & 2.8621 & 1.3539 \\ 0 & 0.6823 & 1.3539 & 1.1400 \end{bmatrix} \tag{12}$$

Estimation

- To estimate the parameter vector, the population analogue of the SMM objective function as

$$J(b) = g(b)' W^* g(b)$$

- The first order condition is

$$\nabla_b \left(g(b)' W^* g(b) \right) = 0 \iff \nabla_b g(b)' W^* g(b) = 0$$

- The derivative of $g(b)$ (an $n \times \ell$ matrix) defined in (8) is:

$$\nabla_b g(b) = -\nabla_b \mu(y(b)) = -\begin{bmatrix} 0 \\ 2b \\ -1 \\ 0 \end{bmatrix} \tag{13}$$

Estimation - cont.

- In that case we can write $\nabla_b g(b)' W^* g(b) = 0$ as

$$-\begin{bmatrix} 0 & 2b & -1 & 0 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 1.1115 & 1.5705 & 0.6823 \\ 0 & 1.5705 & 2.8621 & 1.3539 \\ 0 & 0.6823 & 1.3539 & 1.1400 \end{bmatrix} g(b) = 0$$

- If we evaluate $\nabla_b g(b)$ at $b = b_0$ and compute $\nabla_b g(b_0)' W^*$, then we obtain

$$\left[\begin{array}{cccc} 0 & 0.4590 & 1.2916 & 0.6715 \end{array}\right] g(b_0) = 0 \qquad (14)$$

which is one equation in one unknown b_0.

Estimation - cont.

- What does $\begin{bmatrix} 0 & 0.4590 & 1.2916 & 0.6715 \end{bmatrix} g(b) = 0$ imply?
- Recall this is an overidentified system. Further, recall from (8) that the mean (1) and 2nd order autocovariance (4) do not identify b, the variance (2) does not uniquely identify b, while the 1st order autocovariance (3) uniquely identifies b.
- The most weight is placed on (3) as it should.
- Zero weight is placed on (1) because it is not useful at all for the estimation of b.
- Then why does (4) receive positive weight?
    - Even though the 2nd autocovariance doesn't depend on b, it is correlated with the variance and 1st autocovariance, which is useful for the estimation of b.
    - That is, if we want to make the estimator efficient, we should take the information in the 2nd autocovariance into account.

Computing Standard Errors

- Recall, from Theorem 3.2 of Hansen to compute standard errors we need more than just the optimal weighting matrix.
- Specifically:

$$\sqrt{T}(b_T - b_0) \to N(0, \left[\nabla_b g(b_0)' W^* \nabla_b g(b_0)\right]^{-1})$$

- Hence, we need the derivative of g (an $n \times \ell$ matrix) defined in (8):

$$\begin{aligned} \nabla_b g(b_0) &= -\nabla_b \mu(y(b_0)) \\ &= -\begin{bmatrix} 0 \\ 2b_0 \\ -1 \\ 0 \end{bmatrix} \end{aligned} \tag{15}$$

Computing Standard Errors

- This derivative is also useful to see if the parameter is **Locally Identified**. To see this, take the first order approximation of $g(b)$ around b_0:

$$\begin{aligned} g(b) &\approx g(b_0) + \nabla_b g(b_0)(b - b_0) \\ &= \nabla_b g(b_0)(b - b_0) \end{aligned}$$

  since $g(b_0) = 0$.

- For $b = b_0$ to be the unique solution to $\nabla_b g(b_0)(b - b_0) = 0$, it must be true that

$$\text{rank}(\nabla_b g(b_0)) = \ell$$

- From (15), we can see that $\text{rank}(\nabla_b g(b_0)) = 1 = \ell$, so the parameter is locally identified.
- In contrast, if $\nabla_b g(b_0)$ in (15) was the zero vector (which has rank $0 < \ell$), then the objective would not respond to changes in the parameter.

AGS Sensitivity Matrix Λ

Definition 1. The sensitivity of $\hat{b}$ (a vector of estimated parameters) to $\hat{g}(b_0)$ (model moments) is:

$$\Lambda = -(G'WG)^{-1}G'W$$

where

- G is the Jacobian matrix $\nabla_b g(b_0)$
- W is a weight matrix

Hence Λ acts like the inverse of the Jacobian matrix.

- In the case of minimum distance estimators like SSM (section IV.A.) where $g(b) = M_T(x) - M_N(y(b))$, it measures how variation in model moments impact parameter estimates.
- A big number means the parameter is very sensitive to a given data moment.

Application to MA(1) Case

- Let W be an identity matrix.
- $G = \nabla_b g(b_0) = -\nabla_b \mu(y(b_0)) = -[0, 2b_0, -1, 0]'$ is the Jacobian Matrix.

Then

$$\Lambda = -(G'WG)^{-1} G'W$$

$$= -\left(\begin{bmatrix} 0 & -2b_0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -2b_0 \\ 1 \\ 0 \end{bmatrix} \right)^{-1}$$

$$\times \left(\begin{bmatrix} 0 & -2b_0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)$$

Application to MA(1) Case - cont.

Then

$$\Lambda = \begin{bmatrix} 0 & \frac{2b_0}{4b_0^2+1} & -\frac{1}{4b_0^2+1} & 0 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} & 0 \end{bmatrix}$$

when evaluated at true value of $b_0 = 0.5$.

What does this tell us?

- The estimated parameter b is sensitive to the variance and first order auto-correlation moments.
- If we think that under a different set of modeling assumptions, the estimated variance would be *higher*, we would expect that our estimate of b would also be higher.

Small Samples

- In general we never actually have analytical expressions for $\mu(x)$ and $\mu(y(b))$ so cannot obtain an estimate as above.

- That's why we will use SMM to estimate b in a finite sample.

- With a finite sample of size T data we must construct the $n \times 1$ vector of data moments $M_T(x)$.

- Given the finite sample, in general $M_T(x) \neq \mu(x)$, but $M_T(x) \overset{a.s.}{\to} \mu(x)$ as $T \to \infty$.

True Data Sample

- We first generate a series of random sample $\{\varepsilon\}_{t=1}^{T}$ from $N(0,1)$ and then construct a series of $\{x_t\}_{t=1}^{T}$ using the true DGP in (1) or $x_t = \varepsilon_t - b_0\varepsilon_{t-1}$ with $\varepsilon_0 = 0$, $b_0 = 0.5$, and $T = 200$.

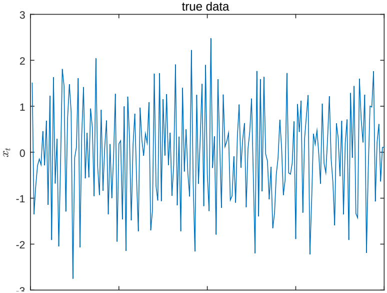


Figure: Simulated 'true' data

Sample moments for the true data

- Using this data, we can compute the $m \times 1$ data moment vector by

$$M_T(x) = \frac{1}{T} \sum_{t=1}^{T} m(x_t). \tag{16}$$

- For our case, the $m = 4$ data moment vector obtained from this simulation is

$$M_T(x) = \begin{bmatrix} -0.0153 \\ 1.1874 \\ -0.4269 \\ -0.0868 \end{bmatrix}. \tag{17}$$

- Recall the population moment matrix is

$$M_T(x) = \begin{bmatrix} 0 \\ 1.25 \\ -0.5 \\ 0 \end{bmatrix}.$$

Sample Var-Covar for the true data

- This data is used to estimate the sample var-covar matrix.
- Unlike the general SMM slides where it was assumed that there was no autocorrelation in $u(x_t, b)$, here we have autocorrelation so apply Newey-West correction in (11).
- Let

$$\widehat{\Gamma}_{T,j} \equiv \frac{1}{T} \sum_{t=j+1}^{T} \left[m(x_t) - M_T(x)\right] \left[m(x_{t-j}) - M_T(x)\right]'$$

  denote the j-th autocovariance of m.
- Then the estimated sample var-covar matrix of $m(x_t)$ is

$$\widehat{S}_{x,T} = \widehat{\Gamma}_{T,0} + \sum_{j=1}^{\infty} \left(1 - \frac{j}{i(T)+1}\right) (\widehat{\Gamma}_{T,j} + \widehat{\Gamma}'_{T,j})$$

  where $i(T)$ is the key to the Newey-West correction (here taken to be 4).

Sample Var-Covar for the true data - cont.

The sample var-covar matrix is given by

$$\hat{S}_{x,T} = \begin{bmatrix} 0.4147 & 0.0058 & -0.0895 & -0.0244 \\ 0.0058 & 1.8946 & -0.8869 & -0.1872 \\ -0.0895 & -0.8869 & 1.2988 & -0.6078 \\ -0.0244 & -0.1872 & -0.6078 & 1.5729 \end{bmatrix} \tag{18}$$

while recall that the population var-covar matrix is given by

$$S_x = \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 4.125 & -2.5 & 0.5 \\ 0 & -2.5 & 2.3125 & -1.25 \\ 0 & 0.5 & -1.25 & 2.0625 \end{bmatrix}$$

SMM Estimation

- Next we consider the model moments.
- We first draw a series of random sample $\{\{e_t^h\}_{t=1}^T\}_{h=1}^H$.
- We will use the same draw in the whole estimation process.
- Given parameter value b, we can compute $\{\{y_t^i(b)\}_{t=1}^T\}_{h=1}^H$ in (2) or

$$y_t^h = e_t^h - be_{t-1}^h$$

where $e_0^h = 0$, $T = 200$, and $H = 10$.

- Then given b, we can compute the simulated model moment

$$M_{TH}(y(b)) = \frac{1}{H}\sum_{i=1}^{H}\frac{1}{T}\sum_{t=1}^{T} m(y_t^h(b)). \tag{19}$$

SMM Estimation - cont.

- Our objective is to choose b so that the weighted sum of squared residuals between the model moments $M_{TH}(y(b))$ in (19) and data moments $M_T(x)$ in (16) is minimized.
- The estimation procedure depends on whether we have the data to form the optimal weighting matrix $\hat{S}_{x,T}^{-1}$ as in (18) or we only have data moments so we cannot directly estimate the var-covar matrix from the data.
- The consistent estimate of the parameter b solves

$$\hat{b}_{TH} = \arg\min_{b} J_{TH}(b) \tag{20}$$

where

$$J_{TH}(b) \equiv [M_T(x) - M_{TH}(y(b))]' W [M_T(x) - M_{TH}(y(b))].$$

SMM Estimation - cont.

The SMM estimate that solves (20) depends on W:

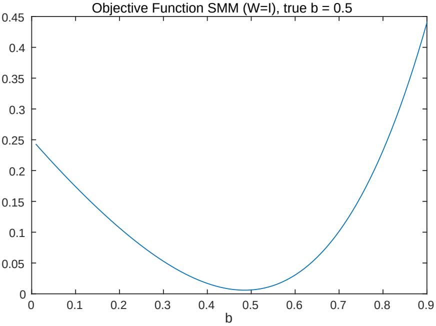
- In the former case where $W^* = \hat{S}_{x,T}^{-1}$, the consistent and efficient estimate of b is $\hat{b}_T^* = 0.4993$.
- In the latter case where we don't have data to estimate $\hat{S}_{x,T}^{-1}$, we can use a two-step procedure.
    - In the first stage where we use $W = I$ second stage, we obtain a consistent estimate of $\hat{b}_{TH}^1 = 0.4850$.
    - We can then use $\hat{b}_{TH}^1$ to generate a model equivalent of $\hat{S}_{y,TH}^{-1}$ to obtain a consistent, efficient estimate $\hat{b}_{y,TH}^2 = 0.4970$.

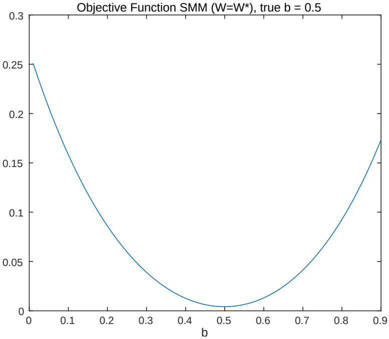
Comparing Data and Model Simulated Weighting Matrices

$$\hat{S}_{x,T}^{-1} = \begin{bmatrix} 2.5146 & 0.2155 & 0.4282 & 0.2301 \\ 0.2155 & 1.0110 & 0.9316 & 0.4836 \\ 0.4282 & 0.9316 & 1.8197 & 0.8206 \\ 0.2301 & 0.4836 & 0.8206 & 1.0140 \end{bmatrix}$$

$$\hat{S}_{y,TH}^{-1} = \begin{bmatrix} 2.2441 & 0.0369 & 0.0023 & 0.0395 \\ 0.0369 & 0.8928 & 1.1272 & 0.4225 \\ 0.0023 & 1.1272 & 2.1335 & 0.9009 \\ 0.0395 & 0.4225 & 0.9009 & 0.9688 \end{bmatrix}.$$

Comparing Data and Model Weighted Sum of Squared Errors





Standard Errors

- Once we have computed the point estimate, we want to compute the standard errors of the estimator.
- In the case of the simulated weighted matrix, we have

$$\sqrt{T}(\widehat{b}^2_{TH} - b_0) \to N(0, (1+1/H)\left[\nabla_b g_T(\widehat{b}^2_{TH})' \widehat{S}^{-1}_{y,TH} \nabla_b g_T(\widehat{b}^2_{TH})\right]^{-1}$$

where

$$\begin{aligned} g_T(b) &\equiv \frac{1}{T}\sum_{t=1}^{T} u(x_t, b) = \frac{1}{T}\sum_{t=1}^{T} m(x_t) - \frac{1}{H}\sum_{h=1}^{H}\frac{1}{T}\sum_{t=1}^{T} m(y_t^h(b)) \\ &= M_T(x) - M_{TH}(y(b)) \end{aligned}$$

- The derivative of g_T is given by

$$\begin{aligned} \nabla_b g_T(\hat{b}^2_{TH}) &= -\nabla_b M_{TH}(y(\hat{b}^2_{TH})) \\ &= -\frac{1}{TH}\sum_{h=1}^{H}\sum_{t=1}^{H}\frac{\partial m(y_t^h(\hat{b}^2_{TH}))}{\partial b} \end{aligned}$$

Standard Errors - cont.

- In general we don't have an analytical formula for this derivative, so we will use the numerical derivative.
- Once can compute $M_{TH}(y(\hat{b}2_{TH}))$, then compute $M_{TH}(y(\hat{b}2_{TH} - s))$, take the difference, and divide by the step size s.
- The result is

$$\frac{\Delta M_{TH}(\hat{b}2_{TH})}{\Delta b} = \begin{bmatrix} -0.0104 \\ 0.9342 \\ -0.9330 \\ -0.0234 \end{bmatrix}$$

- Since there is small sample error, this numerical derivative is broadly consistent with the theoretical result computed in (15) evaluated at $b_0 = 0.5$ given by $[0 \; 1 \;\; -1 \; 0]'$.

Standard Errors - cont.

- The standard error of the $\hat{S}_{y,TH}^{-1}$ weighted estimator is then

$$\text{Std}(\hat{b}_{TH}^2) = \sqrt{\frac{1}{T}\left[\nabla_b g_T(\hat{b}_{TH}^2)' \left\{\left(1+\frac{1}{H}\right)\hat{S}_{y,TH}\right\}^{-1} \nabla_b g_T(\hat{b}_{TH}^2)\right]^{-1}} = 0.089.$$

- The standard error of the $\hat{S}_{x,T}^{-1}$ weighted estimator is

$$\text{Std}(\hat{b}_T^*) = \sqrt{\frac{1}{T}\left[\nabla_b g_T(\hat{b}_T^*)' \left\{\hat{S}_{x,T}\right\}^{-1} \nabla_b g_T(\hat{b}_T^*)\right]^{-1}} = 0.075.$$

- Using $\frac{\Delta M_{TH}(\hat{b}_{TH}^2)}{\Delta b}$ and an identity weighting matrix we can compute the small sample AGS statistic

$$\Lambda = \begin{bmatrix} -0.0060 & 0.5357 & -0.5360 & -0.0134 \end{bmatrix}$$

J-Test

- Once we have estimated the parameter, we can also test if the moment condition is true or not.

$$T\frac{H}{1+H}\times[M_T(x)-M_{TH}(y(\hat{b}^2_{TH}))]'W^*_{TH}[M_T(x)-M_{TH}(y(\hat{b}^2_{TH}))] = 0.7588.$$

- The asymptotic distribution of this test statistics is $\chi(n-k)$, where n is the number of moments $(=4)$ and k is the number of parameters $(=1)$.

- The p value is 0.14, so we cannot reject the hypothesis that the model is true.

Bootstrap

- In order to see the finite sample distribution of the estimators, we can use the bootstrap method. The algorithm is as follows.

  1. Draw ε_t and e_t^h from $N(0,1)$ for $t = 1, 2, \ldots, T$ and $h = 1, 2, \ldots, H$. Compute $(\hat{b}^1_{TH}, \hat{b}^2_{TH,data}, \hat{b}^2_{TH,sim})$ as described.

  2. Repeat 1 using another seed.

- Every time you do step 1, the seed needs to change (which is done automatically by matlab if you don't specify it). Otherwise you will keep getting the same estimators.

Bootstrap - cont.

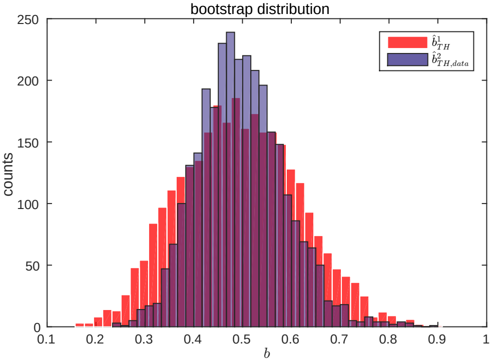


Figure: Bootstrap distributions: histogram

- The histogram of the estimator is plotted in figure 4.
- As theory predicts, $\hat{b}^*_T$, which is the efficient estimator, has a smaller variance than $\hat{b}^1_{y,TH}$.

Bootstrap - cont.

- To make it easier for us to compare the distributions, figure 5 plots the density function of the estimators, obtained by Kernel density estimation

$$f(x) = \frac{1}{\sqrt{2\pi}} \frac{1}{cI} \sum_{i=1}^{I} \exp\left[-\frac{1}{2}\left(\frac{x - x_i}{c}\right)^2\right]$$

  where I is the number of data and c is the bandwidth.
- We can see that the distribution of $\hat{b}^*_{x,T}$ looks very similar to that of $\hat{b}^2_{y,TH}$.
- This is because the model nests the true DGP (in the sense that it is the true DGP at b_0), so even if we use the simulated data to estimate the variance-covariance matrix, we can obtain the efficient estimator.

Bootstrap - cont.

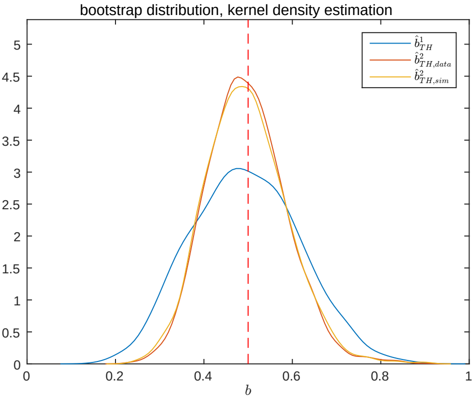


Figure: Bootstrap distributions, approximated by the kernel density estimation.