Efficient Maximum Likelihood Batch Estimation With Pure Time Series Data of a One-Dimensional Cumulative Structural Equation Model

Vytautas Dulskis (presenter), Leonidas Sakalauskas

Vilnius University, Lithuania

September 21, 2022

Structural Equation Modelling (SEM)

- Comprehensive methodology for representing, estimating, and testing a theoretical network of (mostly) linear relations between variables;
- Acknowledged for an explicit assessment of measurement errors, an estimation of latent (unobserved) variables via observed variables, and model testing where a structure can be imposed and assessed as to the fit of the data;
- Combines regression and factor analyses through a measurement model that defines latent variables related to one or more observed variables and a structural model that indicates relationships between latent variables.

Dynamic Structural Equation Model (DSEM)

3.2 Dynamic panel structural equation model

In this section we consider a dynamic panel simultaneous equation model with latent variables and fixed effects (DPSEM(p,q)). A DPSEM(p,q) model for the individual $i = 1, \ldots, N$ at time $t = 1, \ldots, T$ can be written for the generic individual at any time period t using the " t-notation" as

$$\begin{aligned}
\boldsymbol{\eta}_{it} &= \sum_{j=0}^{p} \boldsymbol{B}_j \boldsymbol{\eta}_{it-j} + \sum_{j=0}^{q} \boldsymbol{\Gamma}_j \boldsymbol{\xi}_{it-j} + \boldsymbol{\zeta}_{it} && (3.1) \\
\boldsymbol{y}_{it} &= \boldsymbol{\Lambda}_y \boldsymbol{\eta}_{it} + \boldsymbol{\mu}_{yi} + \boldsymbol{\varepsilon}_{it} && (3.2) \\
\boldsymbol{x}_{it} &= \boldsymbol{\Lambda}_x \boldsymbol{\xi}_{it} + \boldsymbol{\mu}_{xi} + \boldsymbol{\delta}_{it} && (3.3)
\end{aligned}$$

where $\boldsymbol{\eta}_{it} = \left(\eta_{it}^{(1)}, \eta_{it}^{(2)}, \ldots, \eta_{it}^{(m)}\right)'$ and $\boldsymbol{\xi}_{it} = \left(\xi_{it}^{(1)}, \xi_{it}^{(2)}, \ldots, \xi_{it}^{(g)}\right)'$ are vectors of latent variables, $\boldsymbol{y}_{it} = \left((y_{it}^{(1)}, y_{it}^{(2)}, \ldots, y_{it}^{(n)}\right)'$ and $\boldsymbol{x}_{it} = \left(x_{it}^{(1)}, x_{it}^{(2)}, \ldots, x_{it}^{(k)}\right)'$ are vectors of observable variables, and $\boldsymbol{B}_j$ $(m \times m)$, $\boldsymbol{\Gamma}_j$ $(m \times g)$, $\boldsymbol{\Lambda}_x$ $(k \times g)$, and $\boldsymbol{\Lambda}_y$ $(n \times m)$ are coefficient matrices. The contemporaneous and simultaneous coefficients are in $\boldsymbol{B}_0$, and $\boldsymbol{\Gamma}_0$, while $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, $\ldots$, $\boldsymbol{B}_p$, and $\boldsymbol{\Gamma}_1$, $\boldsymbol{\Gamma}_2$, $\ldots$, $\boldsymbol{\Gamma}_q$ contain coefficients of the lagged endogenous and exogenous latent variables. Finally, $\boldsymbol{\mu}_{yi}$ and $\boldsymbol{\mu}_{xi}$ are the $n \times 1$ and $k \times 1$ vectors of individual means, respectively. We treat $\boldsymbol{\mu}_{yi}$ and $\boldsymbol{\mu}_{xi}$ as vectors of coincidental (fixed) parameters, which makes the DPSEM model (3.1)-(3.3) a " fixed-effects" panel model. The statistical assumptions about the variables in (3.1)–(3.3) are as follows.

Cumulative Structural Equation Model (CSEM) Under Consideration (1)

Output measurement equation:

$$y_t = \eta_t + \epsilon_t, \tag{1}$$

where $\{y_t\}$, $t \in \mathbb{Z}^+$ is a sequence of scalar observed outcomes, $\{\eta_t\}$, $t \in \mathbb{Z}^+$ is a sequence of scalar latent spaces, and $\{\epsilon_t\}$, $t \in \mathbb{Z}^+$ is a sequence of independent and identically distributed $\mathcal{N}\left(0, \sigma_y^2\right)$ scalar observed process noises.

Transition equation:

$$\eta_{t+1} = \eta_t + \mu_\eta + L_\eta \xi_{t+1} + \zeta_{t+1}, \qquad \eta_0 = 0, \tag{2}$$

where μ_η is a scalar intercept term, L_η is an $1 \times k$ vector of latent input weights, $\{\xi_{t+1}\}$, $t \in \mathbb{Z}^+$ is a sequence of independent and identically distributed $k \times 1$ vectors of latent input to the latent process (common factors) where $\xi_{t+1} \sim \mathcal{N}\left(0_{k\times 1}, I_{k\times k}\right)$, and $\{\zeta_{t+1}\}$, $t \in \mathbb{Z}^+$ is a sequence of independent and identically distributed $\mathcal{N}\left(0, \sigma_\eta^2\right)$ scalar latent process errors.

Input measurement equation:

$$x_{t+1} = \mu_x + L_x \xi_{t+1} + \delta_{t+1}, \tag{3}$$

where $\{x_{t+1}\}$, $t \in \mathbb{Z}^+$ is a sequence of $m \times 1$ vectors of observed inputs, μ_x is an $m \times 1$ vector of intercept terms, L_x is an $m \times k$ matrix of factor loadings, and $\{\delta_{t+1}\}$, $t \in \mathbb{Z}^+$ is a sequence of independent and identically distributed $m \times 1$ vectors of specific factors where $\delta_{t+1} \sim \mathcal{N}\left(0_{m\times 1}, \operatorname{diag}\left(\sigma_{x_1}^2, .., \sigma_{x_m}^2\right)\right)$.

It is assumed that $\{\xi_{t+1}\}$, $\{\delta_{t+1}\}$, $\{\zeta_{t+1}\}$, and $\{\epsilon_t\}$, $t \in \mathbb{Z}^+$ are mutually independent and $t \leq T$ ($T \in \mathbb{N}$).

CSEM Under Consideration (2)

- We attach μ_η to the end of μ_x and L_η to the end of L_x, and denote the resulting vector and matrix as μ and L, respectively;
- We consider the task of computationally efficient batch maximum likelihood estimation of μ, $LL' + \mathrm{diag}\left(\sigma^2_{x_1}, ..., \sigma^2_{x_m}, \sigma^2_\eta + 2\sigma^2_y\right)$, and σ^2_y;
- The aim is to construct an algorithm that will yield maximum likelihood estimates and be linear in the number of time moments T.

Algorithm Development (1)

Instead of the directly observed output y_t, $t = 0, 1, ..., T$ defined by (1), we consider the differences, as this approach provides a tridiagonal covariance matrix structure that facilitates the solution:

$$\begin{aligned} \Delta y_t &= y_t - y_{t-1} \\ &= \mu_\eta + L_\eta \xi_t + \zeta_t + \left(\epsilon_t - \epsilon_{t-1}\right), \end{aligned} \tag{4}$$

$t = 1, ..., T$.

Algorithm Development (2)

We attach Δy_t to the end of x_t, $t = 1, ..., T$, and denote the resulting vector as z_t, additionally denoting $z = (z_1, ..., z_T)'$. Then

$$\mathsf{E} z_t = \mu$$

and

$$\mathsf{C}_{zz}(\tau) = \begin{cases} LL' + \Psi_0, & \tau = 0 \\ -\Psi_1, & \tau = 1 \\ 0_{(m+1)\times(m+1)}, & otherwise \end{cases},$$

where

$$\Psi_0 = \begin{bmatrix} \sigma_{x_1}^2 & 0 & \dots & 0 & 0 \\ 0 & \sigma_{x_2}^2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \sigma_{x_m}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\eta^2 + 2\sigma_y^2 \end{bmatrix}, \quad \Psi_1 = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_y^2 \end{bmatrix}.$$

Algorithm Development (3)

We denote

$$I = I_{T\times T}, \quad J = J_{T\times T} = \begin{cases} -1, & |i-j| = 1 \\ 0, & otherwise \end{cases},$$

$i = 1, ..., T$, $j = 1, ..., T$. A single covariance matrix containing all the information can be arranged in $(m+1) \times (m+1)$ blocks, each of which contains $T \times T$ elements:

$$\mathsf{C}_{zz} = \Sigma \otimes I + \Psi_1 \otimes J,$$

where $\Sigma = LL' + \Psi_0$.

Algorithm Development (4)

We vectorise the observations and their expectations by defining $z_{vec} = \mathsf{vec}(z)$ and $\mu_{vec} = \mathsf{vec}\left((\mathsf{E}z_1, ..., \mathsf{E}z_T)'\right) = \mathsf{vec}\left((\mu, ..., \mu)'\right)$. Vector z_{vec} has $(m+1)T$–dimensional Gaussian distribution, and, therefore, its probability density is of the form

$$\frac{e^{-\frac{1}{2}d'_{vec}\mathsf{C}_{zz}^{-1}d_{vec}}}{(2\pi)^{\frac{(m+1)T}{2}}\left|\mathsf{C}_{zz}\right|^{\frac{1}{2}}}, \tag{5}$$

where $d_{vec} = z_{vec} - \mu_{vec}$.

Algorithm Development (5)

By taking the logarithm of (5), we obtain the logarithmic likelihood function:

$$\mathcal{L}(\theta) = \mathcal{L}(\theta; z_{vec}) = -\frac{1}{2}\left(\ln(|\mathsf{C}_{zz}|) + d'_{vec}\mathsf{C}_{zz}^{-1}d_{vec} + (m+1)T\ln(2\pi)\right),$$

where $\theta = (\mu_{vec}, \mathsf{C}_{zz})$ are parameters of the multivariate normal distribution.

Since it is true that

$$\arg\max_{\theta}\left[\mathcal{L}(\theta)\right] = \arg\max_{\theta}\left[\frac{\mathcal{L}(\theta)}{T}\right] = \arg\min_{\theta}\left[\frac{\ln(|\mathsf{C}_{zz}|) + d'_{vec}\mathsf{C}_{zz}^{-1}d_{vec}}{T}\right]$$

can be used for both normalisation and simplicity purposes, we further consider the minimisation of function $\mathcal{L}(\theta)$ defined as

$$\mathcal{L}(\theta) = \frac{\ln(|\mathsf{C}_{zz}|) + d'_{vec}\mathsf{C}_{zz}^{-1}d_{vec}}{T}. \tag{6}$$

Algorithm Development (6)

We denote $A = \Sigma \otimes I$, $B = \mathrm{diag}\left(\text{eigenvalues}(J)\right)$, $C = \text{eigenvectors}(J)$, and $D = (\underbrace{0_{T\times T}, ..., 0_{T\times T}}_{m}, C)'$. Then

$$\begin{aligned}
|\mathsf{C}_{zz}| &= \left|A + D\left(\sigma_y^2 B\right) D'\right| \\
&= |A| \left|I + \sigma_y^2 B D' A^{-1} D\right| \\
&= |\Sigma|^T \left|C'C + \sigma_y^2 \left(C'JC\right) \Sigma_l I\right| \\
&= |\Sigma|^T \left|C' \sigma_y^2 \Sigma_l Q C\right| \\
&= |\Sigma|^T \left(\sigma_y^2 \Sigma_l\right)^T \left|C'C\right| |Q| \\
&= \left(|\Sigma|\, \Sigma_l\right)^T \left(\sigma_y^2\right)^T |I|\, |Q| \\
&= |\Sigma_x|^T \left(\sigma_y^2\right)^T |Q| \\
&= \left(|\Sigma_x|\, \sigma_y^2\right)^T |Q|
\end{aligned} \tag{7}$$

Algorithm Development (7)

and

$$
\begin{aligned}
d'_{vec}\mathsf{C}_{zz}^{-1}d_{vec} &= d'_{vec}\left(A + D\left(\sigma_y^2 B\right)D'\right)^{-1} d_{vec} \\
&= d'_{vec}\left(A^{-1} - A^{-1}D\left(\left(\sigma_y^2 B\right)^{-1} + D'A^{-1}D\right)^{-1} D'A^{-1}\right) d_{vec} \\
&= d'_{vec}\left(A^{-1} - A^{-1}DC\left(\left(\sigma_y^2 J\right)^{-1} + \Sigma_l I\right)^{-1} C'D'A^{-1}\right) d_{vec} \\
&= d'_{vec}\left(A^{-1} - \frac{A^{-1}DCJQ^{-1}C'D'A^{-1}}{\Sigma_l}\right) d_{vec} \\
&= d'_{vec}\left(A^{-1} - \frac{A^{-1}DC\left(\frac{1}{\sigma_y^2\Sigma_l}I + J - \frac{1}{\sigma_y^2\Sigma_l}I\right)Q^{-1}C'D'A^{-1}}{\Sigma_l}\right) d_{vec} \\
&= d'_{vec}\left(A^{-1} - \frac{A^{-1}DCQQ^{-1}C'D'A^{-1}}{\Sigma_l} + \frac{A^{-1}DC\left(\frac{1}{\sigma_y^2\Sigma_l}I\right)Q^{-1}C'D'A^{-1}}{\Sigma_l}\right) d_{vec} \\
&= d'_{vec}\left(A^{-1} - \frac{A^{-1}DCC'D'A^{-1}}{\Sigma_l} + \frac{A^{-1}DCQ^{-1}C'D'A^{-1}}{\sigma_y^2\Sigma_l^2}\right) d_{vec} \\
&= \operatorname{tr}\left(d\Sigma^{-1}d'\right) - \frac{(d\Sigma_r)'(d\Sigma_r)}{\Sigma_l} + \frac{(d\Sigma_r)'Q^{-1}(d\Sigma_r)}{\sigma_y^2\Sigma_l^2} \\
&= \operatorname{tr}\left(d_x\Sigma_x^{-1}d'_x\right) + \frac{(d\Sigma_r)'Q^{-1}(d\Sigma_r)}{\sigma_y^2\Sigma_l^2},
\end{aligned}
\tag{8}
$$

where $\Sigma_l = \left[\Sigma^{-1}\right]_{m+1,m+1}$, $\Sigma_r = \left[\Sigma^{-1}\right]_{1:(m+1),m+1}$, $\Sigma_x = L_x L'_x + \mathrm{diag}(\sigma^2_{x_1}, ..., \sigma^2_{x_m})$, $Q = \frac{1}{\sigma_y^2\Sigma_l}I + J$, $d_x = z_x - 1_{T\times 1}\mu'_x$, $d = z - 1_{T\times 1}\mu'$, where $z_x = (x_1, ..., x_T)'$.

Algorithm Development (8)

We take $\theta = \left(s, \Sigma_x, \Sigma_s, \sigma_y^2, \mu_x, \mu_\eta\right)$, here

$$s = e^{-\left(\frac{1}{2}\frac{1}{\sigma_y^2 \Sigma_l}\right)},$$
$$\Sigma_s = \Sigma_x^{-1}\Sigma_{xy},$$

where $\Sigma_{xy} = [\Sigma]_{1:m,m+1}$.
Basically, Σ_s and s are surrogate parameters that replace the direct use of Σ_{xy} and $\Sigma_y = [\Sigma]_{m+1,m+1}$.
We note that $s \in [0, 1]$, and it essentially characterises the weight of σ_y^2 against a combination of both $\sigma_{x_1}^2, ..., \sigma_{x_m}^2, \sigma_\eta^2$ and $[LL']_{m+1,m+1} + \sigma_\eta^2$, with higher values meaning more weight.

Algorithm Development (9)

Σ_{xy} is not among the chosen θ but can be readily calculated through its constituents:

$$\Sigma_{xy} = \Sigma_x \Sigma_s.$$

Likewise, Σ_y has the following expression:

$$\Sigma_y = \frac{\sigma_y^2}{s}(s^2 + 1) + (\Sigma_{xy})' \, \Sigma_s.$$

Algorithm Development (10)

With the aforementioned choice of θ and given expressions (7) and (8), function (6) is ultimately reorganised as follows:

$$\begin{aligned} \mathcal{L}(s, \Sigma_x, \Sigma_s, \sigma_y^2, \mu_x, \mu_\eta) = \frac{1}{T} \Bigg(& \ln \left(\left(|\Sigma_x| \sigma_y^2 \right)^T |Q| \right) \\ & + \operatorname{tr} \left(d_x \Sigma_x^{-1} d_x' \right) \\ & + \frac{1}{\sigma_y^2} \left((d_y - d_x \Sigma_s)' \, Q^{-1} \, (d_y - d_x \Sigma_s) \right) \Bigg), \end{aligned} \tag{9}$$

where $d_y = z_y - 1_{T \times 1} \mu_\eta$, $Q = \left(\frac{s^2+1}{s} \right) I + J$, where $z_y = (\Delta y_1, ..., \Delta y_T)'$.

Algorithm Development (11)

$$\frac{\partial \mathcal{L}}{\partial \Sigma_x} = \Sigma_x^{-1} - \frac{\Sigma_x^{-1} d_x' d_x \Sigma_x^{-1}}{T} = 0_{m\times m} \rightarrow \tag{10}$$

$$\hat{\Sigma}_x = \frac{d_x' d_x}{T}, \tag{11}$$

$$\frac{\partial \mathcal{L}}{\partial \Sigma_s} = \frac{2}{T\sigma_y^2}\left(d_x' Q^{-1} d_x \Sigma_s - d_x' Q^{-1} d_y\right) = 0_{m\times 1} \rightarrow \tag{12}$$

$$\hat{\Sigma}_s = \left(d_x' Q^{-1} d_x\right)^{-1} d_x' Q^{-1} d_y, \tag{13}$$

$$\frac{\partial \mathcal{L}}{\partial \sigma_y^2} = \frac{1}{\sigma_y^2} - \frac{1}{T}\frac{\left(d_y - d_x\Sigma_s\right)' Q^{-1}\left(d_y - d_x\Sigma_s\right)}{\left(\sigma_y^2\right)^2} = 0 \rightarrow \tag{14}$$

$$\hat{\sigma}_y^2 = \frac{\left(d_y - d_x\Sigma_s\right)' Q^{-1}\left(d_y - d_x\Sigma_s\right)}{T}, \tag{15}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mu_x} &= \frac{2}{T\sigma_y^2}\left(\left(\mathbf{1}' Q^{-1}\mathbf{1}\Sigma_s\Sigma_s' + \sigma_y^2 T \Sigma_x^{-1}\right)\mu_x + \left(d_y - z_x\Sigma_s\right)' Q^{-1}\mathbf{1}\Sigma_s - \sigma_y^2\Sigma_x^{-1} z_x'\mathbf{1}\right) \\ &= 0_{m\times 1} \rightarrow\end{aligned} \tag{16}$$

$$\hat{\mu}_x = \left(\mathbf{1}' Q^{-1}\mathbf{1}\Sigma_s\Sigma_s' + \sigma_y^2 T\Sigma_x^{-1}\right)^{-1}\left(\sigma_y^2\Sigma_x^{-1} z_x'\mathbf{1} - \left(d_y - z_x\Sigma_s\right)' Q^{-1}\mathbf{1}\Sigma_s\right), \tag{17}$$

$$\frac{\partial \mathcal{L}}{\partial \mu_\eta} = \frac{2}{T\sigma_y^2}\left(d_x\Sigma_s - d_y\right)' Q^{-1}\mathbf{1} = 0 \rightarrow \tag{18}$$

$$\hat{\mu}_\eta = \frac{\left(z_y - d_x\Sigma_s\right)' Q^{-1}\mathbf{1}}{\mathbf{1}' Q^{-1}\mathbf{1}}, \tag{19}$$

Algorithm Development (12)

where $\mathbf{1} = 1_{T\times 1}$.

- Initial estimators are significantly interdependent, namely $\hat{\Sigma}_x = \hat{\Sigma}_x(\mu_x)$, $\hat{\Sigma}_s = \hat{\Sigma}_s(s, \mu_x, \mu_\eta)$, $\hat{\sigma}_y^2 = \hat{\sigma}_y^2(s, \Sigma_s, \mu_x, \mu_\eta)$, $\hat{\mu}_x = \hat{\mu}_x(s, \Sigma_x, \Sigma_s, \sigma_y^2, \mu_\eta)$, $\hat{\mu}_\eta = \hat{\mu}_\eta(s, \Sigma_s, \mu_x)$;
- We want to obtain the $\hat{\Sigma}_x$, $\hat{\Sigma}_s$, $\hat{\sigma}_y^2$, $\hat{\mu}_x$, $\hat{\mu}_\eta$ that enable explicit (one-off) calculations given s, whose optimal value (i.e. an estimate) is, in turn, obtained by minimising a one-dimensional likelihood function $\tilde{\mathcal{L}}(s)$, thus changing the general problem of function $\mathcal{L}$ (defined by (9)) minimisation of all its variables to obtain the optimal θ (θ^*).

Algorithm Development (13)

By manipulating the derivatives and the estimators obtained from them (10)–(19), we can achieve a gradual way to obtain θ^*:

$$\mu_x^* = \hat{\mu}_x, \tag{20}$$

$$\Sigma_x^* = \hat{\Sigma}_x, \tag{21}$$

$$s^* = \arg\min_{s \in [0,1]} \tilde{\mathcal{L}}(s), \tag{22}$$

$$\Sigma_s^* = \hat{\Sigma}_s\left(s^*\right), \tag{23}$$

$$\mu_\eta^* = \hat{\mu}_\eta\left(s^*\right), \tag{24}$$

$$\sigma_y^{2*} = \hat{\sigma}_y^2\left(s^*\right), \tag{25}$$

Algorithm Development (14)

where

$$\hat{\mu}_x = \frac{z_x' \mathbf{1}}{T}, \tag{26}$$

$$\hat{\Sigma}_x = \frac{\hat{d}_x' \hat{d}_x}{T}, \tag{27}$$

$$\begin{aligned}\hat{\Sigma}_s = &\left(I_{m \times m} - \frac{\left(\hat{d}_x' Q^{-1} \hat{d}_x\right)^{-1} \hat{d}_x' Q^{-1} \mathbf{1} \left(\hat{d}_x' Q^{-1} \mathbf{1}\right)'}{\mathbf{1}' Q^{-1} \mathbf{1}} \right)^{-1} \left(\hat{d}_x' Q^{-1} \hat{d}_x\right)^{-1} \hat{d}_x' Q^{-1} \left(z_y \right. \\ &\left. - \frac{z_y' Q^{-1} \mathbf{1}}{\mathbf{1}' Q^{-1} \mathbf{1}} \mathbf{1} \right),\end{aligned} \tag{28}$$

$$\hat{\mu}_\eta = \frac{\left(z_y - \hat{d}_x \hat{\Sigma}_s\right)' Q^{-1} \mathbf{1}}{\mathbf{1}' Q^{-1} \mathbf{1}}, \tag{29}$$

$$\hat{\sigma}_y^2 = \frac{\left(\hat{d}_y - \hat{d}_x \hat{\Sigma}_s\right)' Q^{-1} \left(\hat{d}_y - \hat{d}_x \hat{\Sigma}_s\right)}{T}, \tag{30}$$

$$\tilde{\mathcal{L}} = \frac{\ln\left(|Q|\right)}{T} + \ln\left(\hat{\sigma}_y^2\right), \tag{31}$$

where $\hat{d}_x = z_x - \mathbf{1} \hat{\mu}_x'$, $\hat{d}_y = z_y - \mathbf{1} \hat{\mu}_\eta$.

Algorithm Development (15)

To be indeed efficient with (20)–(25), we need to efficiently invert the $T \times T$ matrix Q in expressions of the form $u'Q^{-1}v$ (u and v are generally $T \times 1$ vectors or $T \times m$ matrices; however, here we limit the calculations to atomic elements only, as they make $u'Q^{-1}v$, whether it is a scalar, vector, or matrix; i.e., we assume that u and v are both $T \times 1$ vectors) as well as calculate its determinant.

Algorithm Development (16)

Let us denote $c_T = \frac{1}{T}\frac{\left([u]_{1:T,1}\right)'\left([Q]_{1:T,1:T}\right)^{-1}[v]_{1:T,1}}{q_T}$; here $q_T = \left[\left([Q]_{1:T,1:T}\right)^{-1}\right]_{T,T}$. Thus, instead of calculating terms of the form $u'Q^{-1}v$ in (28)–(31), we calculate terms of the form c_T since $\frac{1}{T}\frac{1}{q_T}$ simply cancels out in $\hat{\Sigma}_s$ and $\hat{\mu}_\eta$, while, for $\hat{\sigma}_y^2$ and $\tilde{\mathcal{L}}$, the use of c_T introduces robust expressions, as we now calculate them as follows:

$$\begin{aligned}\hat{\sigma}_y^2 &= q_T g \\ &= s\frac{f_T}{f_{T+1}}g\end{aligned}$$

and

$$\tilde{\mathcal{L}} = h + \ln(g),$$

where

$$\begin{aligned}h &= \frac{\ln(|Q|)}{T} + \ln q_T \\ &= \ln\left(\frac{f_T}{\left(f_{T+1}\right)^{\left(1-\frac{1}{T}\right)}}\right)\end{aligned}$$

and

$$g = \frac{\hat{\sigma}_y^2}{q_T},$$

where g is c_T with $u = v = \hat{d}_y - \hat{d}_x\hat{\Sigma}_s$ and $f_T = \sum_{i=0}^{T-1} s^{2i}$ is to be calculated recursively:

$$f_T = f_{T-1} + s^{2(T-1)}.$$

Algorithm Development (17)

Recursively, c_T is calculated as follows:

$$c_T = \left(1 - \frac{1}{T}\right) \frac{f_{T-1} f_{T+1}}{f_T^2} c_{T-1} + \frac{a_T b_T}{T},$$

where $a_T = \dfrac{\left([u]_{1:T,1}\right)' \left[\left([Q]_{1:T,1:T}\right)^{-1}\right]_{1:T,T}}{q_T}$ and $b_T = \dfrac{\left([v]_{1:T,1}\right)' \left[\left([Q]_{1:T,1:T}\right)^{-1}\right]_{1:T,T}}{q_T}$ are also to be calculated recursively:

$$a_T = s \frac{f_{T-1}}{f_T} a_{T-1} + [u]_{T,1},$$

$$b_T = s \frac{f_{T-1}}{f_T} b_{T-1} + [v]_{T,1}.$$

The starting values for all the recursions, namely a_1, b_1, c_1, and f_1, are

$$a_1 = [u]_{1,1} \,,$$

$$b_1 = [v]_{1,1} \,,$$

$$c_1 = [u]_{1,1} \, [v]_{1,1} \,,$$

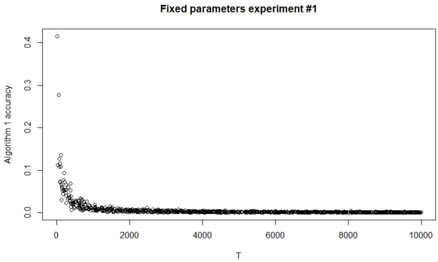
$$f_1 = 1.$$

Experimental Results (1)

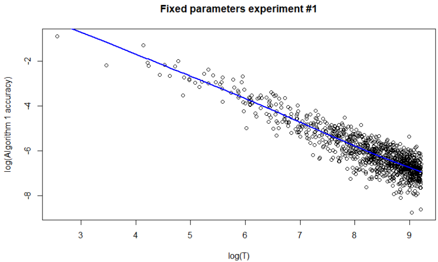
- Fixed parameter experiments for numerical verification of convergence;
- Fixed sample size experiments for comparing against a competing method;
- The estimation accuracy is measured by

$$\frac{\ln\left(|\mathsf{C}_{zz}|\right) - \ln\left(\left|\hat{\mathsf{C}}_{zz}\right|\right) + \operatorname{tr}\left(\hat{\mathsf{C}}_{zz}\mathsf{C}_{zz}^{-1}\right) + \left(\hat{\mu}_{vec} - \mu_{vec}\right)' \mathsf{C}_{zz}^{-1}\left(\hat{\mu}_{vec} - \mu_{vec}\right) - (m+1)T}{T}.$$

Experimental Results (2)



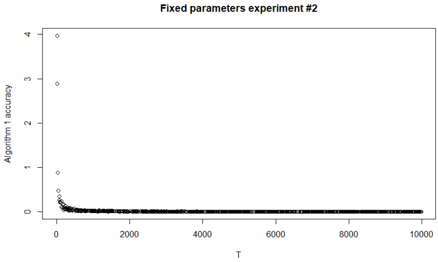
(a) Linear scale



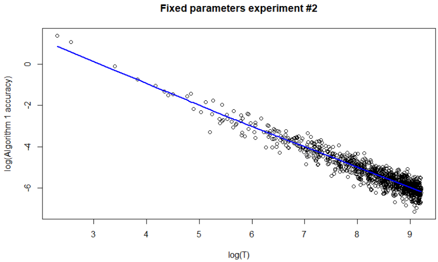
(b) Logarithmic scale

A fixed parameters experiment with $m = 2$, $k = 1$, $L = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$, $\sigma = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$, $\mu = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.

Experimental Results (3)



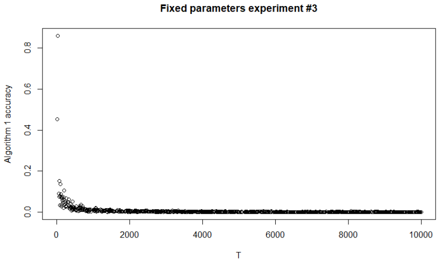
(a) Linear scale



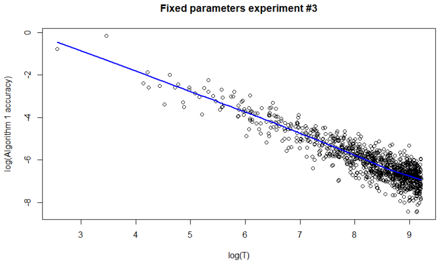
(b) Logarithmic scale

A fixed parameters experiment with $m = 4$, $k = 2$, $L = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}$, $\sigma = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$, $\mu = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$.

Experimental Results (4)



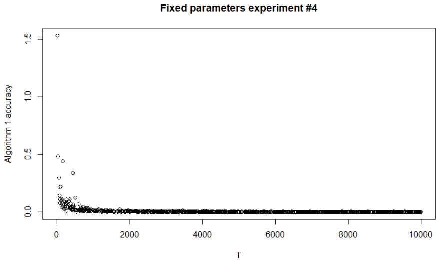
(a) Linear scale



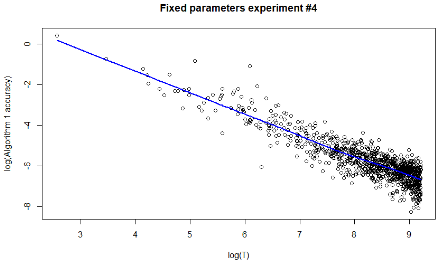
(b) Logarithmic scale

A fixed parameters experiment with $m = 2$, $k = 1$, $L = \begin{pmatrix} 3.033113 \\ 1.354755 \\ -7.729820 \end{pmatrix}$, $\sigma = \begin{pmatrix} 5.9592531 \\ 3.5804998 \\ 4.2880942 \\ 0.5190332 \end{pmatrix}$, $\mu = \begin{pmatrix} -4.716447 \\ -2.024185 \\ 6.722683 \end{pmatrix}$ $(s = 0.00437782)$.

Experimental Results (5)



(a) Linear scale

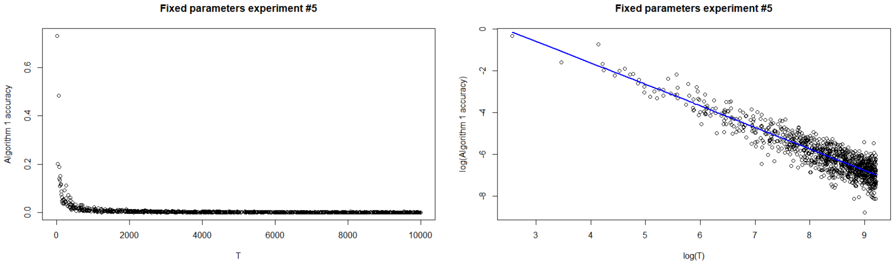


(b) Logarithmic scale

A fixed parameters experiment with $m = 2$, $k = 1$, $L = \begin{pmatrix} -2.178953 \\ -2.828553 \\ -2.551128 \end{pmatrix}$, $\sigma = \begin{pmatrix} 4.431249655 \\ 0.011324964 \\ 0.008721256 \\ 5.194705613 \end{pmatrix}$,

$\mu = \begin{pmatrix} 2.7596023 \\ 2.9909256 \\ -0.7072652 \end{pmatrix}$ $(s = 0.9974178)$.

Experimental Results (6)

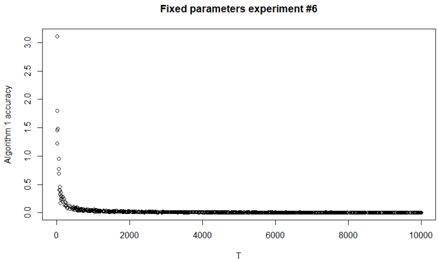


(a) Linear scale (b) Logarithmic scale

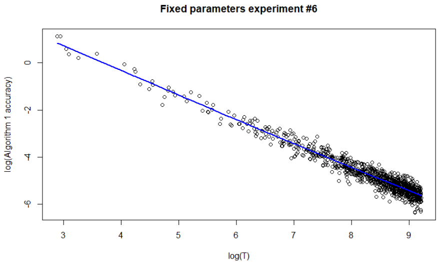
A fixed parameters experiment with $m = 2$, $k = 1$, $L = \begin{pmatrix} 8.054983 \\ -1.189202 \\ 5.106852 \end{pmatrix}$, $\sigma = \begin{pmatrix} 4.3582932 \\ 0.5084394 \\ 1.1895825 \\ 2.8840829 \end{pmatrix}$, $\mu = \begin{pmatrix} -9.280441 \\ -2.191468 \\ 4.389951 \end{pmatrix}$

$(s = 0.5042926)$.

Experimental Results (7)



(a) Linear scale

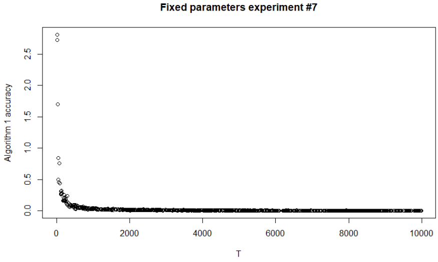


(b) Logarithmic scale

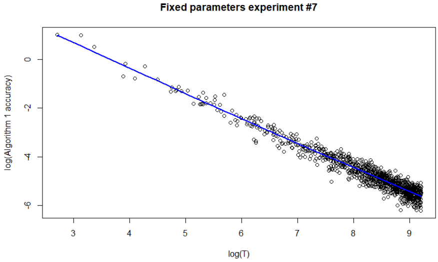
A fixed parameters experiment with $m = 6$, $k = 6$, $L = \begin{pmatrix} -5.566499 & -6.114174 & 3.047041 & -1.505944 & -2.793123 & -2.5011761 \\ 5.813173 & -7.042847 & -9.433562 & -1.540058 & -8.379231 & 1.6832248 \\ 6.726056 & 9.266465 & 7.952709 & -2.436595 & -4.732580 & -4.2417507 \\ -4.111609 & -9.024494 & 9.936109 & 9.479690 & 3.747947 & -1.0394100 \\ -4.857664 & 1.638594 & 9.163533 & 5.084826 & 4.927076 & -3.5418603 \\ 9.343039 & 9.740263 & -2.936545 & -8.826716 & 8.084022 & 0.5527933 \\ -9.694904 & -8.954491 & -2.958250 & 2.983702 & -1.353331 & 7.7284050 \end{pmatrix}$, $\sigma = \begin{pmatrix} 2.56916160 \\ 8.49587681 \\ 2.57191516 \\ 1.62583368 \\ 0.73195778 \\ 7.51540902 \\ 0.09016039 \\ 9.07366179 \end{pmatrix}$,

$\mu = \begin{pmatrix} -0.5337413 \\ 1.9107166 \\ 2.8861898 \\ -1.6599453 \\ -2.6935648 \\ -0.8379153 \\ -4.5070887 \end{pmatrix}$ $(s = 0.495446)$.

Experimental Results (8)



(a) Linear scale

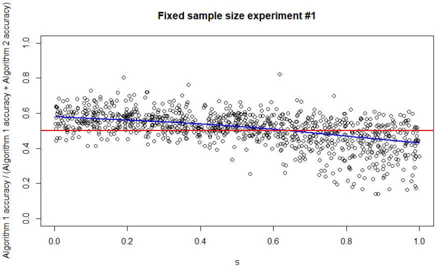


(b) Logarithmic scale

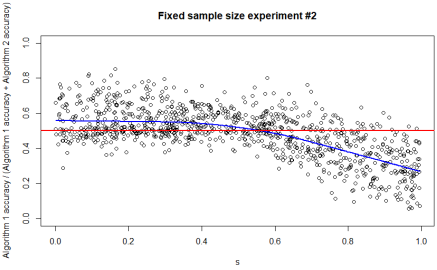
A fixed parameters experiment with $m = 6$, $k = 1$, $L = \begin{pmatrix} -9.403870 \\ -4.837518 \\ 7.215641 \\ 1.767852 \\ 1.195050 \\ -2.918301 \\ 6.156870 \end{pmatrix}$, $\sigma = \begin{pmatrix} 4.737127 \\ 2.706422 \\ 5.006524 \\ 4.268941 \\ 2.671212 \\ 5.720389 \\ 5.894513 \\ 8.741166 \end{pmatrix}$, $\mu = \begin{pmatrix} 9.764480 \\ -7.753982 \\ -3.484666 \\ -2.827956 \\ -1.503179 \\ -6.081425 \\ -5.228186 \end{pmatrix}$

$(s = 0.4998399)$.

Experimental Results (9)

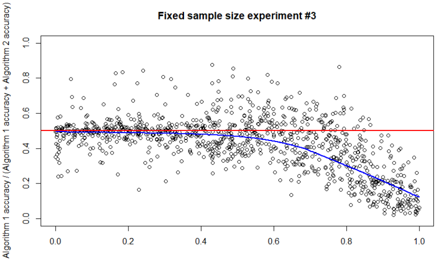


A fixed sample size experiment with $T = 10$.

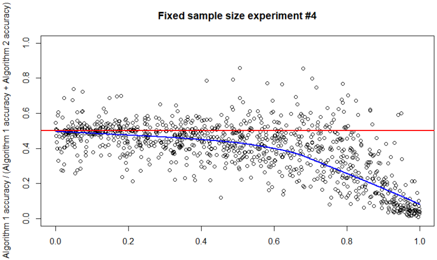


A fixed sample size experiment with $T = 20$.

Experimental Results (10)

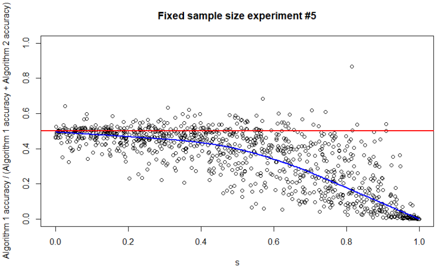


A fixed sample size experiment with $T = 50$.

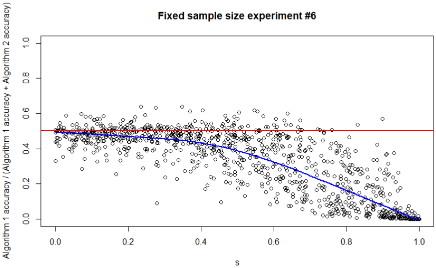


A fixed sample size experiment with $T = 100$.

Experimental Results (11)



A fixed sample size experiment with $T = 1000$.



A fixed sample size experiment with $T = 10000$.

Implementation in R

- EMLI – Efficient Maximum Likelihood Inference for Linear Dynamical Models (https://CRAN.R-project.org/package=EMLI);
- Three functions available that allow generating the considered CSEM data, running the developed algorithm, and evaluating the estimation accuracy;
- More models and features to be added.

Final Remarks

- Positive experimental results, yet theoretical results are needed as well;
- Recursive nature paves the way for online estimation algorithms;
- Efficiently solving special cases contributes to the overall arsenal of options that might also lead to insights when considering more general models.