Estimating State-Space Models in Innovations Form using the Expectation Maximisation Algorithm

Adrian Wills*, Thomas B. Schön+ and Brett Ninness*

*School of Electrical Engineering and Computer Science, University of Newcastle, Newcastle, Australia.
+Division of Automatic Control, Linköping University, Linköping, Sweden.

Supported by the Australian Research Council, the Swedish foundation for strategic research in the center MOVIII and the Swedish Research Council in the Linnaeus center CADICS.
The main motivation for this work is not to show how to identify innovation form models.

**Main motivation:** Provide insight into how to derive an EM algorithm.

More specifically, provide an explicit example of the fact that we should be careful when the "missing" data is selected.
How can we estimate models on innovation form,

\[ x_{t+1} = Ax_t + Bu_t + Ke_t, \quad e_t \sim \mathcal{N}(0, R), \]
\[ y_t = Cx_t + Du_t + e_t, \quad x_1 \sim \mathcal{N}(\mu, P_1), \]

based on the information in the measured input-output responses

\[ U_N \triangleq [u_1, \cdots, u_N], \quad Y_N \triangleq [y_1, \cdots, y_N], \]

using the maximum likelihood (ML) framework,

\[ \hat{\theta} = \arg \max_{\theta} p_\theta(y_1, \cdots, y_N), \]
A state-space model on innovations form can also be written as

\[y_t = \hat{y}_{t|t-1} + e_t,\]
\[\hat{y}_{t|t-1} = Cx_t + Du_t,\]
\[x_{t+1} = (A - KC)x_t + (B - KD)u_t + Ky_t,\]

where

\[\hat{y}_{t|t-1} \triangleq E\{y_t \mid Y_{t-1}\}\]

The (well-known) key observation is that the state is a deterministic function of the observations \(y_t, u_t\) and the initial state \(x_1\).
A state-space model on innovations form can also be written as

\[ y_t = \hat{y}_{t|t-1} + e_t, \]
\[ \hat{y}_{t|t-1} = Cx_t + Du_t, \]
\[ x_{t+1} = (A - KC)x_t + (B - KD)u_t + Ky_t, \]

where

\[ \hat{y}_{t|t-1} \triangleq E\{y_t \mid Y_{t-1}\} \]

The (well-known) key observation is that the state is a deterministic function of the observations \( y_t, u_t \) and the initial state \( x_1 \).
The EM algorithm provides an iterative solution to the ML problem

$$\hat{\theta} = \arg \max_{\theta} p_{\theta}(y_1, \cdots, y_N) = \arg \max_{\theta} \log p_{\theta}(Y_N)$$

EM is concerned with the joint log-likelihood

$$L_{\theta}(Y_N, Z) = \log p_{\theta}(Y_N, Z)$$

of the measurements $Y_N$ and the "missing" data $Z$.

**Key** design variable: The "missing" data $Z$. 

Adrian Wills, Thomas B. Schön and Brett Ninness

Estimating State-Space Models in Innovations Form using the Expectation Maximisation Algorithm
Algorithm (Expectation Maximisation (EM))

1. Set $k = 0$ and initialize $\theta_0$ such that $L_{\theta_0}(Y_N)$ is finite.

2. **Expectation (E) step:** Compute
   
   $$Q(\theta, \theta_k) = E_{\theta_k} \left\{ \log p_\theta(Z, Y_N) \mid Y_N \right\}.$$ 

3. **Maximisation (M) step:** Compute
   
   $$\theta_{k+1} = \arg \max_\theta Q(\theta, \theta_k).$$ 

4. **If not converged, update $k := k + 1$ and return to step 2.**
Most EM algorithms for identification of state-space models are based on (starting with Shumway & Stoffer (1982))

\[ Z = \{x_1, \cdots, x_N\}. \]

This will not work for models on innovation form, since it results in \( p_\theta(Y_N, Z) \) having a Dirac-delta form.

**Main contribution:** Establishing that by choosing the missing data as the unobserved initial value, \( Z = x_1 \) we obtain a good EM algorithm.
Compute the $Q$-function,

$$Q(\theta, \theta_k) = -\log \det P_1 - N \log \det R$$

$$- \operatorname{Tr} \left\{ P_1^{-1} \left( (\hat{x}_{1|N} - \mu)(\hat{x}_{1|N} - \mu)^T + P_{1|N} \right) \right\}$$

$$- \operatorname{Tr} \left\{ R^{-1} \sum_{t=1}^{N} \varepsilon_t \varepsilon_t^T \right\} - \operatorname{Tr} \left\{ R^{-1} \sum_{t=1}^{N} C P_t C^T \right\}$$

where

$$\hat{x}_{1|N} \triangleq \operatorname{E}_{\theta_k} \left\{ x_1 \mid Y_N \right\},$$

$$\varepsilon_t \triangleq y_t - \hat{y}_{t|t-1},$$

$$P_t \triangleq \operatorname{Cov}_{\theta_k} \left\{ x_t \mid Y_{t-1} \right\}$$

$$P_{1|N} \triangleq \operatorname{Cov}_{\theta_k} \left\{ x_1 \mid Y_N \right\}$$

$$\hat{y}_{t|t-1} = \operatorname{E}_{\theta_k} \left\{ y_t \mid Y_{t-1} \right\}$$
Partition $\theta$ according to

$$\theta = [\eta^T, \beta^T]^T,$$

$\eta$ parametrizes $P_1, R$ and $\mu$, and $\beta$ parametrizes $A, B, C, D, K$.

Finding $\eta$ is straightforward, but there are now closed form expressions for $\beta$, which is found by maximizing

$$\tilde{Q}(\beta, \theta_k) = - \log \det \left( \frac{1}{N} \sum_{t=1}^{N} \epsilon_t \epsilon_t^T + CP_tC^T \right).$$

which is done using a quasi-Newton (BFGS) method.
Resulting Algorithm

Algorithm (EM for identifying innovation models)

1. Set \( k = 0 \) and initialize \( \theta_0 \) such that \( L_{\theta_0}(Y_N) \) is finite.

2. **Expectation (E) step:** Based on \( \theta_k \) run a Kalman smoother to obtain \( \hat{x}_{1|N} \) and \( P_{1|N} \).

3. **Maximisation (M) step:** Use a quasi-Newton search algorithm to maximise \( Q(\theta_{k+1}, \theta_k) \) over \( \theta_{k+1} \).

4. If not converged, update \( k := k + 1 \) and return to step 2.
Consider the following output error (OE) model,

\[ x_{t+1} = ax_t + bu_t, \quad x_1 \sim \mathcal{N}(\mu, p_1), \]
\[ y_t = x_t + du_t + e_t, \quad e_t \sim \mathcal{N}(0, r). \]

with the following true parameters,

\[ \theta^* = [a^* \quad b^* \quad d^* \quad r^* \quad \mu^* \quad p_1^*]^T = [0.5 \quad 1 \quad 0 \quad 0.1 \quad 0 \quad 1]^T. \]

- \( N = 100 \) samples were used.
- Input signal given by \( u_t \sim \mathcal{N}(0, 1) \).
- Initial parameter guess taken as \( \theta_0 \in \mathcal{U}[0, 1] \).
- Monte Carlo simulation with \( M = 100 \) runs.
Consider the following output error (OE) model,

\[ x_{t+1} = ax_t + bu_t, \quad x_1 \sim \mathcal{N}(\mu, p_1), \]
\[ y_t = x_t + du_t + e_t, \quad e_t \sim \mathcal{N}(0, r). \]

with the following true parameters,

\[ \theta^* = [a^* \ b^* \ d^* \ r^* \ \mu^* \ p_1^*]^T = [0.5 \ 1 \ 0 \ 0.1 \ 0 \ 1]^T. \]

- \( N = 100 \) samples were used.
- Input signal given by \( u_t \sim \mathcal{N}(0, 1) \).
- Initial parameter guess taken as \( \theta_0 \in \mathcal{U}[0, 1] \).
- Monte Carlo simulation with \( M = 100 \) runs.
Numerical Example (OE) - Results

Failed runs:

<table>
<thead>
<tr>
<th>DGS</th>
<th>EM</th>
<th>EM+DGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>
Same setup as before, but an innovation model

$$x_{t+1} = \begin{pmatrix} \theta_1 & 1 \\ 0 & \theta_2 \end{pmatrix} x_t + \begin{pmatrix} \theta_3 \\ \theta_4 \end{pmatrix} u_t + \begin{pmatrix} \theta_6 \\ \theta_7 \end{pmatrix} e_t,$$

$$y_t = \begin{pmatrix} 1 & 1 \end{pmatrix} x_t + \theta_5 u_t + e_t.$$

$$x_1 \sim \mathcal{N} \begin{pmatrix} \begin{pmatrix} \theta_9 \\ \theta_{10} \end{pmatrix}, \begin{pmatrix} \theta_{11} & 0 \\ \theta_{12} & \theta_{13} \end{pmatrix} \begin{pmatrix} \theta_{11} & 0 \\ \theta_{12} & \theta_{13} \end{pmatrix}^T \end{pmatrix},$$

$$e_t \sim \mathcal{N}(0, \theta_8^2).$$
Failed runs:

<table>
<thead>
<tr>
<th>DGS</th>
<th>EM</th>
<th>EM+DGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>70%</td>
<td>1%</td>
<td>0%</td>
</tr>
</tbody>
</table>
Conclusions and Future Work

- **Main contribution:** Establishing that by choosing the missing data as the initial value, \( Z = x_1 \) we obtain a good EM algorithm.

- This illustrates that you have to be careful when choosing the "missing" data \( Z \).

Interesting question: Does this tell us anything about the case with singular \( P \) for general state-space models?

Mixed use of EM and gradient based search for ML (more this afternoon, 13.50-14.10 in Room 209).
Main contribution: Establishing that by choosing the missing data as the initial value, $Z = x_1$ we obtain a good EM algorithm.

This illustrates that you have to be careful when choosing the "missing" data $Z$.

Interesting question: Does this tell us anything about the case with singular $P$ for general state-space models?

Mixed use of EM and gradient based search for ML (more this afternoon, 13.50-14.10 in Room 209).