Time Series and Sequence Learning

Lecture 6 – Learning of State Space Models

Johan Alenlöv, Linköping University

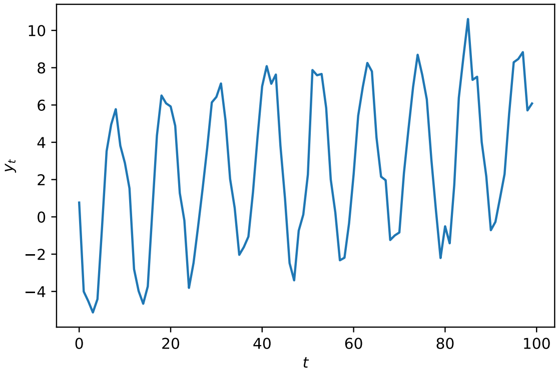
2024-09-17

Information

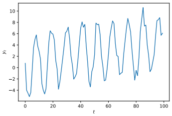
I have been made aware that there is an issue with exporting to pdf on the computers.

Use convert to html instead and submit that together with your jupyter-notebook.

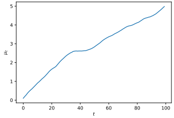
Summary: Structural time series



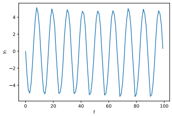
Summary: Structural time series



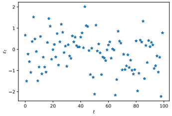
=



+



+



$$y_t = \mu_t + \gamma_t + \varepsilon_t$$

Summary: Trend component

A $k-1$th order polynomial trend model $\Delta^k \mu_t = \zeta_t$ can be written as

$$\alpha_t = \begin{bmatrix} c_1 & c_2 & \cdots & c_{k-1} & c_k \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \alpha_{t-1} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix} \zeta_t,$$

$$\mu_t = \begin{bmatrix} 1 & 0 & 0 & \vdots & 0 \end{bmatrix} \alpha_t,$$

where the state vector is

$$\alpha_t = \begin{bmatrix} \mu_t & \mu_{t-1} & \cdots & \mu_{t-k+1} \end{bmatrix}^\top$$

and $c_i = (-1)^{i+1} \binom{k}{i}$.

Summary: Seasonal component

A s period seasonal model, $\sum_{j=0}^{s-1} \gamma_{t-j} = \omega_t$, can be written a

$$\alpha_t = \begin{bmatrix} -1 & -1 & \cdots & -1 & -1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \alpha_{t-1} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \omega_t$$

$$\gamma_t = \begin{bmatrix} 1 & 0 & 0 & \vdots & 0 \end{bmatrix} \alpha_t,$$

where the state vector is

$$\alpha_t = \begin{bmatrix} \gamma_t & \gamma_{t-1} & \cdots & \gamma_{t-s+2} \end{bmatrix}^T.$$

Summary: Structural time series

A general structural time series model

$$y_t = \mu_t + \gamma_t + \varepsilon_t$$

can be written in state space form using block matrices.

State vector: $\alpha_t = \begin{bmatrix} \mu_t & \mu_{t-1} & \cdots & \mu_{t-k+1} & \gamma_t & \gamma_{t-1} & \cdots & \gamma_{t-s+2} \end{bmatrix}^\mathsf{T}$

State space model:

$$\begin{aligned} \alpha_t &= \begin{bmatrix} T_{[\mu]} & \\ & T_{[\gamma]} \end{bmatrix} \alpha_{t-1} + \begin{bmatrix} R_{[\mu]} & \\ & R_{[\gamma]} \end{bmatrix} \eta_t, \quad && \eta_t \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_\zeta^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix} \right), \\ y_t &= \begin{bmatrix} Z_{[\mu]} & Z_{[\gamma]} \end{bmatrix} \alpha_t + \varepsilon_t, && \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2). \end{aligned}$$

Summary: The Kalman filter

For any s, t, denote by $\hat{\alpha}_{t|s} = \mathbb{E}[\alpha_t \,|\, y_{1:s}]$ and $P_{t|s} = \mathrm{Cov}(\alpha_t \,|\, y_{1:s})$.

Thm. For an LGSS model, $p(\alpha_t \,|\, y_{1:s}) = \mathcal{N}(\alpha_t \,|\, \hat{\alpha}_{t|s}, P_{t|s})$.

Of particular interest are:

- Filtering distribution,

$$p(\alpha_t \,|\, y_{1:t}) = \mathcal{N}(\alpha_t \,|\, \hat{\alpha}_{t|t}, P_{t|t}).$$

- (1-step) Predictive distributions,

$$p(\alpha_t \,|\, y_{1:t-1}) = \mathcal{N}(\alpha_t \,|\, \hat{\alpha}_{t|t-1}, P_{t|t-1}),$$
$$p(y_t \,|\, y_{1:t-1}) = \mathcal{N}(y_t \,|\, \hat{y}_{t|t-1}, F_{t|t-1}).$$

Summary: ARMA model in state space form

State space formulation of ARMA: Consider the ARMA(p, q) model,

$$y_t = \sum_{j=1}^{p} a_j y_{t-j} + \sum_{j=1}^{q} b_j \eta_{t-j} + \eta_t.$$

Let $d = \max(p, q+1)$ and define $a_j = 0$ for $j > p$ and $b_j = 0$ for $j > q$. Then, an equivalent **state space form** is given by

$$\alpha_t = \begin{bmatrix} a_1 & a_2 & \cdots & a_{d-1} & a_d \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \alpha_{t-1} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \eta_t,$$

$$y_t = \begin{bmatrix} 1 & b_1 & \cdots & b_{d-2} & b_{d-1} \end{bmatrix} \alpha_t$$

Summary: Stability of state space model

Thm. A state space model of dimension $d = \dim(\alpha_t)$ is:

- Stable iff $|\lambda_j| < 1, \quad j = 1, \ldots, d,$
- Marginally stable iff $|\lambda_j| \leq 1, \quad j = 1, \ldots, d,$
- Unstable iff $|\lambda_j| > 1$ for any $j = 1, \ldots, d,$

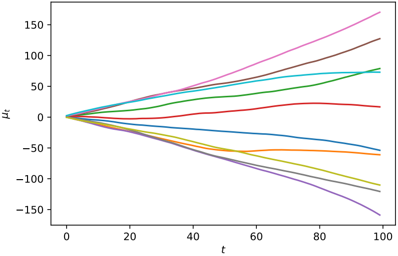
where λ_j, $j = 1, \ldots, d$ are the eigenvalues of T.

Summary: ex) Eigenvalues of linear trend model

Linear trend model:

$$\alpha_t = \overbrace{\begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}}^{F} \alpha_{t-1} + \overbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}^{R} \zeta_t$$

$$\mu_t = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{Z} \alpha_t$$



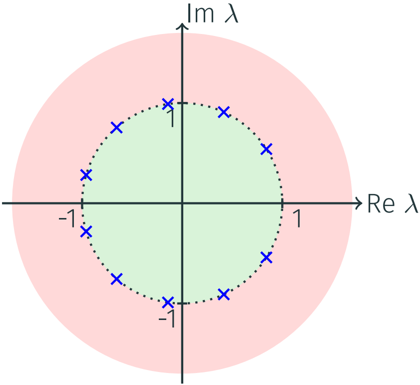
Check for stability by computing the eigenvalues

$$\mathrm{eig}(F) \Rightarrow \lambda_1 = \lambda_2 = 1.$$

The trend model is marginally stable!

Seasonal model:

$$T = \begin{bmatrix} -1 & -1 & \cdots & -1 & -1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$



The seasonal model is marginally stable!

Summary: Stability of structural time series

The structural time series models that we have proposed are

designed to be marginally stable!

Marginal stability results in desirable properties:

- Real eigenvalues $\lambda_j = 1 \Rightarrow$ polynomial drift/trend.
- Complex eigenvalues with $|\lambda_j| = 1 \Rightarrow$ periodicity/seasonality.

Likelihood estimation

Recap: The log-likelihood of the local level model

The Local Level Model given by

$$
\begin{aligned}
y_t &= \mu_t + \varepsilon_t, & \varepsilon_t &\overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2) \\
\mu_{t+1} &= \mu_t + \eta_{t+1}, & \eta_{t+1} &\overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\eta^2)
\end{aligned}
$$

and initial distribution $\mu_1 \sim \mathcal{N}(a_1, P_1)$

Log-Likelihood Given a sequence $y_{1:n}$ the log-likelihood is given by

$$
\begin{aligned}
\ell(\theta) &= \log p_\theta(y_1) + \sum_{t=2}^{n} \log p_\theta(y_t \mid y_{1:t-1}) \\
&= -\frac{n}{2} \log 2\pi - \frac{1}{2} \sum_{t=1}^{n} \left(\log F_{t \mid t-1}(\theta) + \frac{(y_t - \hat{\mu}_{t \mid t-1}(\theta))^2}{F_{t \mid t-1}(\theta)} \right)
\end{aligned}
$$

Def. A Linear Gaussian State-Space (LGSS) model is given by:

$$\begin{aligned} \alpha_t &= T\alpha_{t-1} + R\eta_t, & \eta_t &\sim \mathcal{N}(0, Q), \\ y_t &= Z\alpha_t + \varepsilon_t & \varepsilon_t &\sim \mathcal{N}(0, \sigma_\epsilon^2), \end{aligned}$$

and initial distribution $\alpha_1 \sim \mathcal{N}(a_1, P_1)$.

As in the local level model we write,

$$\ell(\theta) = \log p_\theta(y_1) + \sum_{t=2}^{n} \log p_\theta(y_t \mid y_{1:t-1})$$

Remains to find the distribution $y_t \mid y_{1:t-1}$.

Calculating the log-likelihood

We again need to look at the distribution of $y_t \mid y_{1:t-1}$.

Gaussian distribution $\Rightarrow$ find mean and variance.

$$\begin{aligned}
\mathbb{E}[y_t \mid y_{1:t-1}] &= \mathbb{E}[Z\alpha_t + \varepsilon_t \mid y_{1:t-1}] = Z\hat{\alpha}_{t|t-1} = \hat{y}_{t|t-1} \\
\mathrm{Var}[y_t \mid y_{1:t-1}] &= \mathrm{Var}[Z\alpha_t + \varepsilon_t \mid y_{1:t-1}] \\
&= Z\,\mathrm{Var}[\alpha_t \mid y_{1:t-1}]Z^\top + \sigma_\epsilon^2 \\
&= ZP_{t|t-1}Z^\top + \sigma_\epsilon^2 = F_{t|t-1}
\end{aligned}$$

This gives us

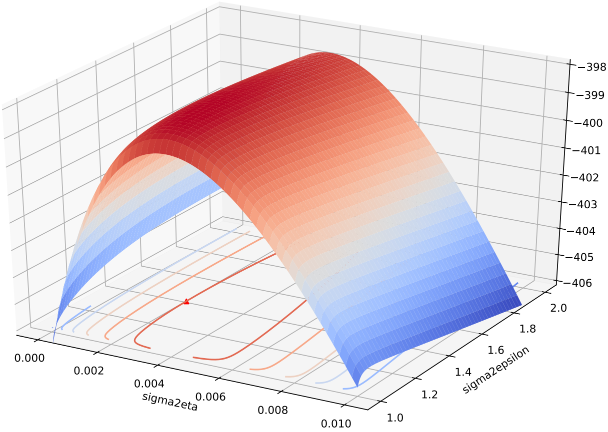
$$\log p_{\theta}(y_t \mid y_{1:t-1}) = \mathrm{const} - \frac{1}{2}\left(\log|F_{t|t-1}| + (y_t - \hat{y}_{t|t-1})^\top F_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})\right)$$

and

$$\ell(\theta) = \mathrm{const} - \frac{1}{2}\sum_{t=1}^{n}\left(\log|F_{t|t-1}| + (y_t - \hat{y}_{t|t-1})^\top F_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})\right)$$

ex) Maximum likelihood in local level model

We return to the example of lecture 4.



Maximum found for $\sigma_\varepsilon^2 = 1.37$ and $\sigma_\eta^2 = 0.002$.

Expectation-Maximization

Another approach to parameter estimation

- Calculating the derivatives of $\ell(\theta)$ is a hard problem.
- If we had access to $\alpha_{1:n}$ it would be much easier.

$$\log p_{\theta}(\alpha_{1:n}, y_{1:n}) = \text{const.} - \frac{1}{2}[n \log |\sigma_{\epsilon}^2| + (n-1) \log |Q| + \sum_{i=1}^{n} (y_i - Z\alpha_i)^{\mathsf{T}} \sigma_{\varepsilon}^{-2} (y_i - Z\alpha_i) + \sum_{i=2}^{n} (\alpha_i - T\alpha_{i-1})^{\mathsf{T}} R Q^{-1} R^{\mathsf{T}} (\alpha_i - T\alpha_{i-1})]$$

$$\log p_{\theta}(\alpha_{1:n}, y_{1:n}) = \text{const.} - \frac{1}{2}[n \log |\sigma_{\epsilon}^2| + (n-1) \log |Q| + \sum_{i=1}^{n} \varepsilon_i^{\mathsf{T}} \sigma_{\varepsilon}^{-2} \varepsilon_i + \sum_{i=2}^{n} \eta_i^{\mathsf{T}} Q^{-1} \eta_i]$$

$$\varepsilon_i = y_i - Z\alpha_i$$

$$\eta_i = R^{\mathsf{T}}(\alpha_i - T\alpha_{i-1})$$

- Easy to take derivatives of this and maximize.

Expectation-Maximization

In the Expectation Maximization (EM) algorithm we alternate two steps,

1. E-step: Calculate $\mathcal{Q}(\theta, \tilde{\theta}) = \mathbb{E}[\log p_\theta(\alpha_{1:n}, y_{1:n}) \mid y_{1:n}, \tilde{\theta}]$
2. M-step: Find $\theta^\star$ that maximizes $\mathcal{Q}(\theta, \tilde{\theta})$.

We have that, (using $\mathbb{E}[x^\top A x] = \mathrm{tr}[A\Sigma] + m^\top A m$ when $x \sim \mathcal{N}(m, \Sigma)$)

$$\mathbb{E}[\log p_\theta(\alpha_{1:n}, y_{1:n}) \mid y_{1:n}, \tilde{\theta}] = \text{const.} - \frac{1}{2} \sum_{t=1}^{n} \Big[\log|\sigma_\epsilon^2| + \log|Q|$$

$$+ \{\hat{\varepsilon}_{t|n}^2 + \mathrm{Var}[\varepsilon_t \mid y_{1:n}]\}\sigma_\varepsilon^{-2} + \mathrm{tr}[\{\hat{\eta}_{t|n}\hat{\eta}_{t|n}^\top + \mathrm{Var}[\eta_t \mid y_{1:n}]\}Q^{-1}]\Big],$$

where $\hat{\varepsilon}_{t|n}, \mathrm{Var}[\varepsilon_t \mid y_{1:n}], \hat{\eta}_{t|n}$, and $\mathrm{Var}[\eta_t \mid y_{1:n}]$ are the smoothed mean and variances of ε_t and η_t.

To find $\theta^\star$ maximize $\mathcal{Q}(\theta, \tilde{\theta})$ by taking the derivative and set the derivative to zero.

One Slide on the Proof

The Smoothing Distribution

The smoothing distribution

- State smoothing refers to the problem of estimating $\alpha_t \mid y_{1:n}$ for $t < n$.
- Often separated into three classes:
    - Fixed-interval smoothing, when n is fixed.
    - Fixed-point smoothing, when t is fixed and $n = t + 1, t + 2, \ldots$.
    - Fixed-lag smoothing, when $t = n - \ell$.
- In our case we are interested in the distributions

$$\eta_t \mid y_{1:n} \qquad \varepsilon_t \mid y_{1:n},$$

  this is known as disturbance smoothing.
- In the LGSS model the distributions will be Gaussian.

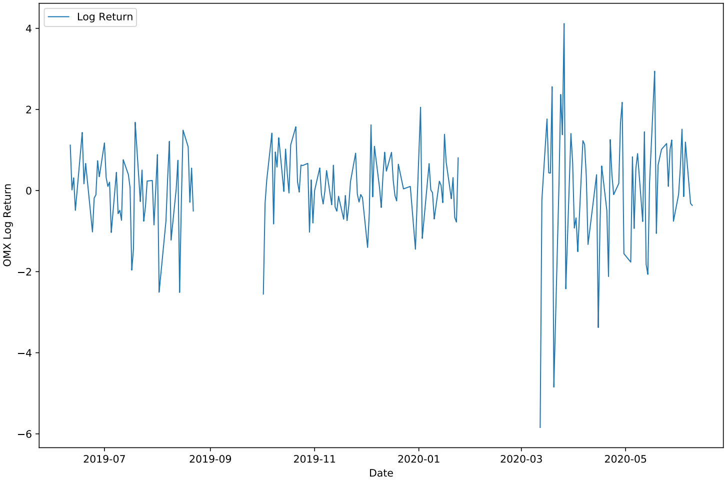
State smoothing

- Smoothing is typically a two-step algorithm.
    1. A filter is run in the forward direction ($t = 1, 2, \ldots, n$)
    2. A smoother is run in the backward direction ($t = n, n-1, \ldots, 1$)
- During the backward pass we will "correct" the filter distributions to the smoothing distributions.
- For $\hat{\alpha}_{t|n}$ we get,

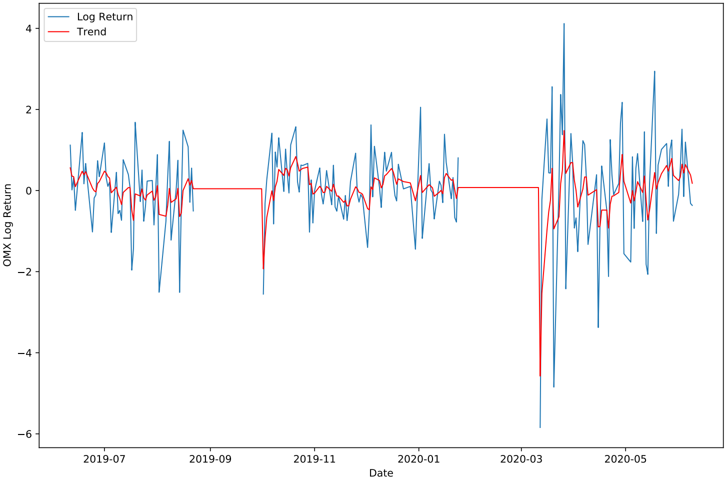
$$\begin{aligned} L_t &= T - TK_tZ \\ r_{t-1} &= Z^{\intercal} F_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1}) + L_t^{\intercal} r_t \\ \hat{\alpha}_{t|n} &= \hat{\alpha}_{t|t-1} + P_{t|t-1} r_t. \end{aligned}$$

Initialized using $r_n = 0$

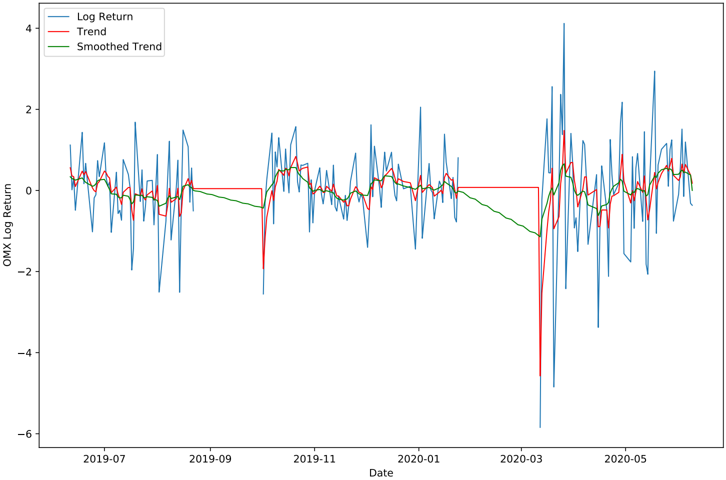
ex) Filter vs. Smoothing distribution



ex) Filter vs. Smoothing distribution



ex) Filter vs. Smoothing distribution



Kalman filter:

1. **Initialize:** Set $\hat{\alpha}_{1|0} = a_1$ and $P_{1|0} = P_1$.
2. **for** $t = 1, 2, \ldots$
   (a) **Measurement update:** // Skip if y_t is unavailable

   · Predict y_t:
   $$\begin{cases} \hat{y}_{t|t-1} = Z\hat{\alpha}_{t|t-1}, \\ F_{t|t-1} = ZP_{t|t-1}Z^{\mathsf{T}} + \sigma_\varepsilon^2 \end{cases}$$

   · Kalman gain:
   $$K_t = P_{t|t-1}Z^{\mathsf{T}}F_{t|t-1}^{-1}$$

   · Update filter:
   $$\begin{cases} \hat{\alpha}_{t|t} = \hat{\alpha}_{t|t-1} + K_t(y_t - \hat{y}_{t|t-1}), \\ P_{t|t} = (I - K_tZ)P_{t|t-1} \end{cases}$$

   (b) **Measurement update:**

   · Predict α_t:
   $$\begin{cases} \hat{\alpha}_{t+1|t} = T\hat{\alpha}_{t|t}, \\ P_{t+1|t} = TP_{t|t}T^{\mathsf{T}} + RQR^{\mathsf{T}} \end{cases}$$

State smoothing for general SSM

State smoother:

1. **Initialize:** Run the **Kalman Filter** and save the Kalman gains and the predictive distributions.
2. **Initialize:** Set $r_n = 0$ and $N_n = 0$
3. **for** $t = n, n-1, \ldots 1$

- Calculate:
$$\begin{cases} L_t = T - TK_tZ \\ r_{t-1} = Z^\top F_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1}) + L_t^\top r_t \\ N_{t-1} = Z^\top F_{t|t-1}^{-1} Z + L_t^\top N_t L_t \end{cases}$$

- State smoothing:
$$\begin{cases} \hat{\alpha}_{t|n} = \hat{\alpha}_{t|t-1} + P_{t|t-1} r_{t-1} \\ P_{t|n} = P_{t|t-1} - P_{t|t-1} N_{t-1} P_{t|t-1} \end{cases}$$

Disturbance smoothing for general SSM

Disturbance smoother:

1. **Initialize:** Run the **Kalman Filter** and save the Kalman gains and the predictive distributions.
2. **Initialize:** Set $r_n = 0$ and $N_n = 0$
3. **for** $t = n, n-1, \ldots 1$

- Calculate:

$$\begin{cases} C_t = T^\top N_t T \\ D_t = F_{t|t-1}^{-1} + K_t^\top C_t K_t \\ u_t = F_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1}) - K_t^\top T^\top r_t \end{cases}$$

- Observation noise:

$$\begin{cases} \hat{\varepsilon}_{t|n} = \sigma_\epsilon^2 u_t \\ \mathrm{Var}[\varepsilon_t \mid y_{1:n}] = \sigma_\epsilon^2 - \sigma_\epsilon^2 D_t \sigma_\epsilon^2 \end{cases}$$

- State noise:

$$\begin{cases} \hat{\eta}_{t|n} = QR^\top r_t \\ \mathrm{Var}[\eta_t \mid y_{1:n}] = Q - QR^\top N_t R Q \end{cases}$$

- Time update:

$$\begin{cases} r_{t-1} = Z^\top u_t + T^\top r_t \\ N_{t-1} = Z^\top D_t Z + C_t - Z^\top K_t^\top C_t - C_t K_t Z \end{cases}$$

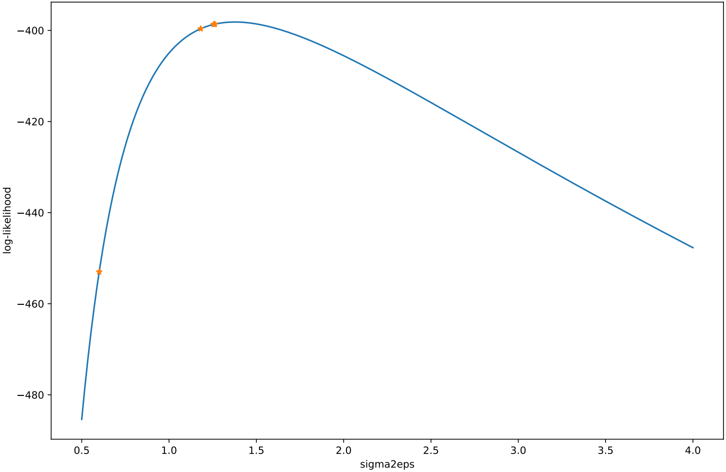
The EM algorithm

- Set an initial parameter value θ_0.
- For $k = 0, 1, \ldots$ do:
    1. Calculate the smoothing distribution using the **disturbance smoother** with the current parameter value θ_k.
    2. Set $\theta_{k+1} = \operatorname{arg\,max} \mathcal{Q}(\theta, \theta_k)$.

  Until convergence.

ex) The EM algorithm

Let's look at σ^2_ε for the local level model.



Summary

A few concepts to summarize lecture 6:

- **Stability:** Results in a stationary state process (possibly after an initial transient). Corresponds to all eigenvalues of T being strictly within the unit circle.
- **Marginal stability:** Results in a state process that grows polynomially and/or shows a non-diminishing periodic pattern. Corresponds to some eigenvalues of T being *on* the unit circle.
- **Log-likelihood:** The log-likelihood for a LGSS can be calculated using the Kalman filter.
- **Expectation-Maximization:** Algorithm for maximum likelihood estimation. Iterates two steps, the E-step and M-step.
- **State Smoothing:** When estimating the hidden state α_t conditioned on data $y_{1:n}$ for $n > t$.
- **Disturbance Smoothing:** Estimation of the *noise variables* η_t and ε_t conditioned on $y_{1:n}$ for $n > t$.