# Time Series Analysis

Week 12 –

Peder Bacher
Department of Applied Mathematics and Computer Science
Technical University of Denmark

April 26, 2024

# Week 12: Outline of the lecture

Recursive and adaptive estimation:

- Recursive LS, Section 11.1
- Recursive pseudo-linear regression, Section 11.2
- Model-based adaptive estimation, Section 11.4

# Why recursive and adaptive estimation?

- As time passes we get more information
- New information should be included by “adjustment” rather than recalculating everything
- Models are approximations
- The best approximation may change over time
- Makes it possible to produce software which learns as new data becomes available

# RLS – Types of models considered

REG:

$$Y_t = \mu + \beta_1 U_{1,t} + \beta_2 U_{2,t} + \ldots + \beta_m U_{m,t} + \varepsilon_t$$

AR:

$$\phi(B) Y_t = \mu + \varepsilon_t \Leftrightarrow$$

$$Y_t = \mu - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \ldots - \phi_p Y_{t-p} + \varepsilon_t$$

FIR:

$$Y_t = \mu + \omega(B) U_t + \varepsilon_t$$

$$= \mu + \omega_0 U_t + \omega_1 U_{t-1} + + \ldots \omega_s U_{t-s} + \varepsilon_t$$

ARX:

$$\phi(B) Y_t = \mu + \omega(B) U_t + \varepsilon_t \Leftrightarrow$$

$$Y_t = \mu - \phi_1 Y_{t-1} - \ldots - \phi_p Y_{t-p} + \omega_0 U_t + \ldots + \omega_s U_{t-s} + \varepsilon_t$$

# Generic form of the models considered

$$\begin{aligned} Y_t &= \boldsymbol{x}_t^T \boldsymbol{\theta} + \varepsilon_t \\ &= \theta_1 x_{1,t} + \theta_2 x_{2,t} + \ldots + \theta_\ell x_{\ell,t} + \varepsilon_t \end{aligned}$$

Example:

$$Y_t = \mu \cdot \underbrace{1}_{x_{1,t}} + \phi_2 \cdot \underbrace{(-Y_{t-2})}_{x_{2,t}} + \omega_1 \cdot \underbrace{U_{t-1}}_{x_{3,t}} + \varepsilon_t$$

# LS-estimate at time $t$

Model:

$$Y_t = \boldsymbol{x}_t^T \boldsymbol{\theta} + \varepsilon_t$$

Data ($\boldsymbol{x}$ may contain lagged values of the "real" input/output):

$$Y_1, Y_2, Y_3, Y_4, \ldots, Y_{t-1}, Y_t$$

$$\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_4, \boldsymbol{x}_4, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t$$

LS-estimate based on $t$ observations:

$$S_t(\boldsymbol{\theta}) = \sum_{s=1}^{t} (Y_s - \boldsymbol{x}_s^T \boldsymbol{\theta})^2$$

$$\widehat{\boldsymbol{\theta}}_t = \arg\min_{\theta} S_t(\boldsymbol{\theta}) = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{Y}$$

# From one time step to the next (in an easy way)

The trick is to realize that:

$$\boldsymbol{R}_t = \boldsymbol{X}^T \boldsymbol{X} = \boldsymbol{x}_1 \boldsymbol{x}_1^T + \boldsymbol{x}_2 \boldsymbol{x}_2^T + \ldots + \boldsymbol{x}_t \boldsymbol{x}_t^T = \sum_{s=1}^{t} \boldsymbol{x}_s \boldsymbol{x}_s^T = \sum_{s=1}^{t-1} \boldsymbol{x}_s \boldsymbol{x}_s^T + \boldsymbol{x}_t \boldsymbol{x}_t^T$$

$$\boldsymbol{h}_t = \boldsymbol{X}^T \boldsymbol{Y} = \boldsymbol{x}_1 Y_1 + \boldsymbol{x}_2 Y_2 + \ldots + \boldsymbol{x}_t Y_t = \sum_{s=1}^{t} \boldsymbol{x}_s Y_s = \sum_{s=1}^{t-1} \boldsymbol{x}_s Y_s + \boldsymbol{x}_t Y_s$$

Where:

$$\boldsymbol{x}_t \boldsymbol{x}_t^T = \begin{bmatrix} x_{1,t} x_{1,t} & x_{1,t} x_{2,t} & \cdots & x_{1,t} x_{\ell,t} \\ x_{2,t} x_{1,t} & x_{2,t} x_{2,t} & \cdots & x_{2,t} x_{\ell,t} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\ell,t} x_{1,t} & x_{\ell,t} x_{2,t} & \cdots & x_{\ell,t} x_{\ell,t} \end{bmatrix} \qquad \boldsymbol{x}_t Y_t = \begin{bmatrix} x_{1,t} Y_t \\ x_{2,t} Y_t \\ \vdots \\ x_{\ell,t} Y_t \end{bmatrix}$$

# The RLS algorithm

$$\widehat{\boldsymbol{\theta}}_t = \boldsymbol{R}_t^{-1}\boldsymbol{h}_t$$

$$\boldsymbol{R}_t = \sum_{s=1}^{t} \boldsymbol{x}_s \boldsymbol{x}_s^{t} = \boldsymbol{x}_t \boldsymbol{x}_t^{T} + \sum_{s=1}^{t-1} \boldsymbol{x}_s \boldsymbol{x}_s^{T} = \underline{\boldsymbol{x}_t \boldsymbol{x}_t^{T} + \boldsymbol{R}_{t-1}}$$

$$\boldsymbol{h}_t = \sum_{s=1}^{t} \boldsymbol{x}_s Y_s = \boldsymbol{x}_t Y_t + \sum_{s=1}^{t-1} \boldsymbol{x}_s Y_s = \underline{\boldsymbol{x}_t Y_t + \boldsymbol{h}_{t-1}}$$

Initialization:

- $\boldsymbol{R}_0 = \mathbf{0}$ (matrix of zeros)
- $\boldsymbol{h}_0 = \mathbf{0}$ (vector of zeros)
- Start estimating $\widehat{\boldsymbol{\theta}}_t$ when $\boldsymbol{R}_t$ is invertible

# The RLS algorithm – 2 equivalent formulations

1. :

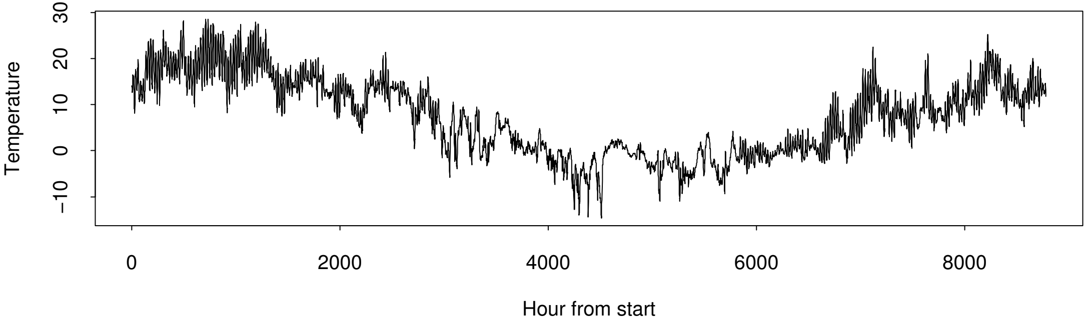
$$\boldsymbol{R}_t = \boldsymbol{R}_{t-1} + \boldsymbol{x}_t \boldsymbol{x}_t^T$$
$$\widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_{t-1} + \boldsymbol{R}_t^{-1} \boldsymbol{x}_t (Y_t - \boldsymbol{x}_t^T \widehat{\boldsymbol{\theta}}_{t-1})$$

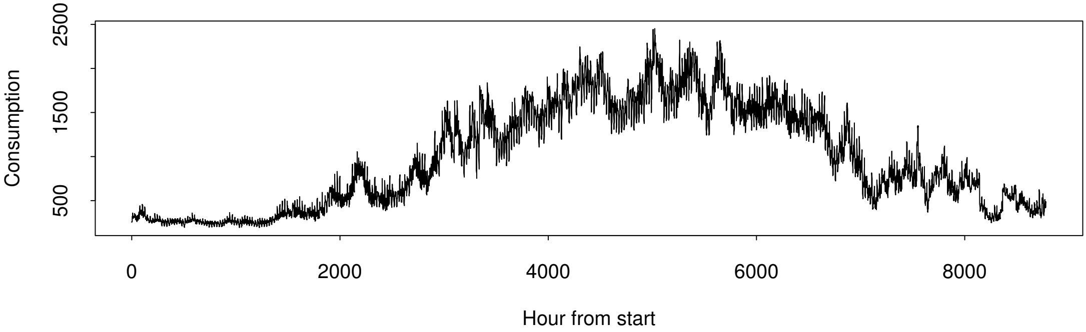
2. Where $\boldsymbol{P}_t = \boldsymbol{R}_t^{-1}$:

$$\boldsymbol{P}_t = \boldsymbol{P}_{t-1} - \frac{\boldsymbol{P}_{t-1} \boldsymbol{x}_t}{1 + \boldsymbol{x}_t^T \boldsymbol{P}_{t-1} \boldsymbol{x}_t} \boldsymbol{x}_t^T \boldsymbol{P}_{t-1}$$
$$\boldsymbol{K}_t = \frac{\boldsymbol{P}_{t-1} \boldsymbol{x}_t}{1 + \boldsymbol{x}_t^T \boldsymbol{P}_{t-1} \boldsymbol{x}_t}$$
$$\widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_{t-1} + \boldsymbol{K}_t (Y_t - \boldsymbol{x}_t^T \widehat{\boldsymbol{\theta}}_{t-1})$$

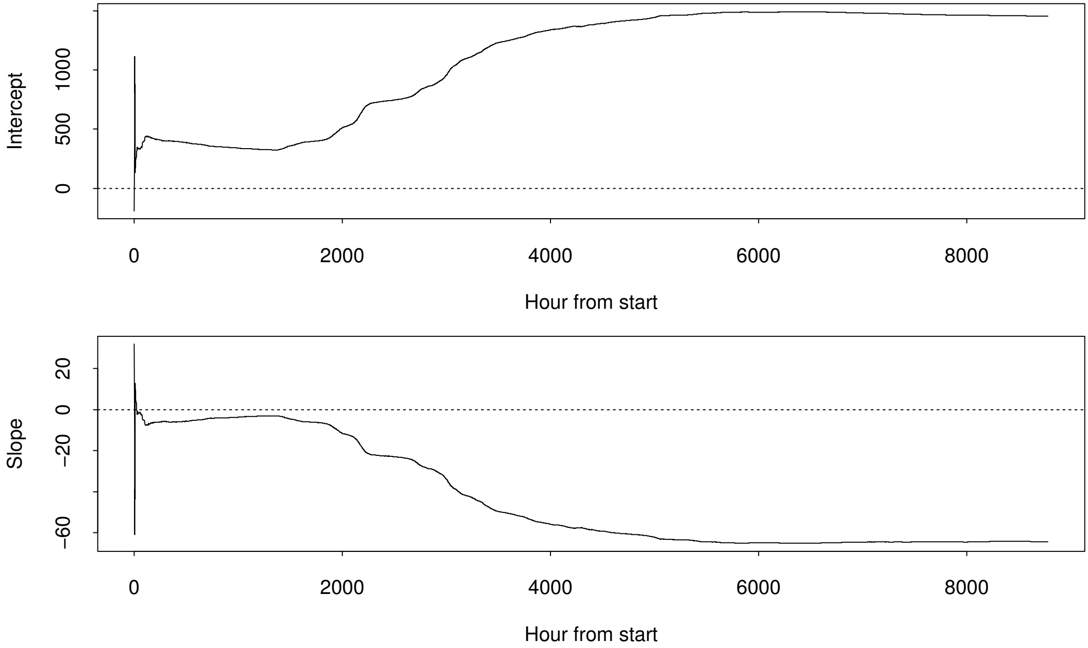
Which one would you prefer for applications and why? Second expression avoids matrix-inversion.

# Example: $HC_t = \mu + \theta_1 T_t + \varepsilon_t$



# Example: $HC_t = \mu + \theta_1 T_t + \varepsilon_t$



# Forgetting old observations

- So far we have a way of updating the estimates as the data set grows
- What if the underlying dynamics change over time? We forget old observations:

$$\widehat{\boldsymbol{\theta}}_t = \arg\min_{\theta} S_t(\boldsymbol{\theta}) = (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{Y}$$

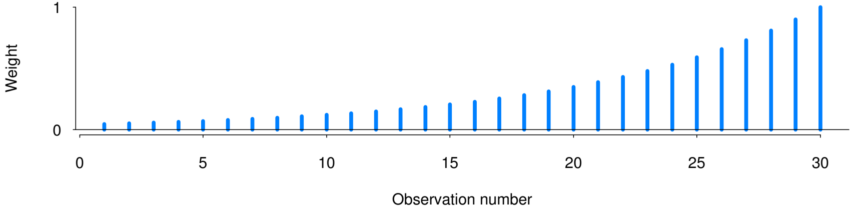
$$S_t(\boldsymbol{\theta}) = \sum_{s=1}^{t} \beta(t, s)(Y_s - \boldsymbol{x}_s^T \boldsymbol{\theta})^2$$

  where $\boldsymbol{W} = \text{diag}(\beta(t, 1), \beta(t, 2), \ldots, \beta(t, t-1), 1)$
- $\beta(t, s)$ expresses how we assign weights to old observations. What does $\beta(t, s)$ usually look like?

# Exponential decay of weights

- Let's first consider $\beta(t, s) = \lambda^{t-s}$ $(0 < \lambda \leq 1)$

$$\lambda = 1: \text{ What we did with the previous algorithms}$$
$$0 < \lambda < 1: \text{ We "forget" in an exponential manner}$$




- In the general case it turns out that if the sequence of weights can be written

$$\beta(t, s) = \lambda(t)\beta(t-1, s) \qquad 1 \leq s \leq t-1$$
$$\beta(t, t) = 1$$

Then the estimates can be updated recursively. Interpret the limitations these equations impose.

# The Adaptive Recursive LS algorithm

$$\begin{aligned} \boldsymbol{R}_t &= \boldsymbol{x}_t \boldsymbol{x}_t^T + \lambda(t) \boldsymbol{R}_{t-1} \\ \boldsymbol{h}_t &= \boldsymbol{x}_t Y_t + \lambda(t) \boldsymbol{h}_{t-1} \\ \widehat{\boldsymbol{\theta}}_t &= \boldsymbol{R}_t^{-1} \boldsymbol{h}_t \end{aligned}$$

# The Adaptive RLS algorithm – 2 equivalent formulations

1. Eliminating $\boldsymbol{h}_t$:

$$\boldsymbol{R}_t = \lambda(t)\boldsymbol{R}_{t-1} + \boldsymbol{x}_t\boldsymbol{x}_t^T$$
$$\widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_{t-1} + \boldsymbol{R}_t^{-1}\boldsymbol{x}_t(Y_t - \boldsymbol{x}_t^T\widehat{\boldsymbol{\theta}}_{t-1})$$

2. Eliminating $\boldsymbol{h}_t$ and avoiding matrix-inversion:

$$\boldsymbol{K}_t = \frac{\boldsymbol{P}_{t-1}\boldsymbol{x}_t}{\lambda(t) + \boldsymbol{x}_t^T\boldsymbol{P}_{t-1}\boldsymbol{x}_t}$$
$$\widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_{t-1} + \boldsymbol{K}_t(Y_t - \boldsymbol{x}_t^T\widehat{\boldsymbol{\theta}}_{t-1})$$
$$\boldsymbol{P}_t = \frac{1}{\lambda(t)}\left(\boldsymbol{P}_{t-1} - \frac{\boldsymbol{P}_{t-1}\boldsymbol{x}_t\boldsymbol{x}_t^T\boldsymbol{P}_{t-1}}{\lambda(t) + \boldsymbol{x}_t^T\boldsymbol{P}_{t-1}\boldsymbol{x}_t}\right)$$

# Constant forgetting

- If $\lambda(t) = \lambda$ we call $\lambda$ the *forgetting factor* and define the *memory* as

$$T_0 = \sum_{i=0}^{\infty} \lambda^i = 1 + \lambda + \lambda^2 + \lambda^3 + \lambda^4 + \ldots = \frac{1}{1-\lambda}$$

- Given a data set an optimal value of $\lambda$ can be found by “trial and error”
- Instead of a linear grid of $\lambda$'s, use a linear grid of $T_0$'s
- Important detail: The initial estimates might be poor. How should this be taken into account when choosing $\lambda$?

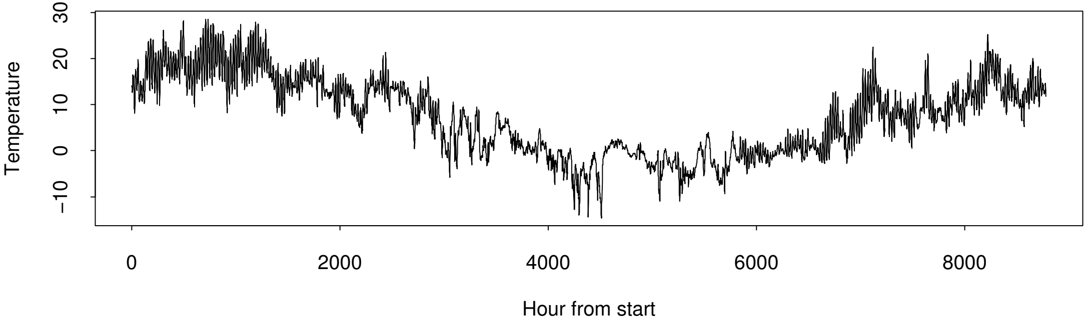
# Variable forgetting

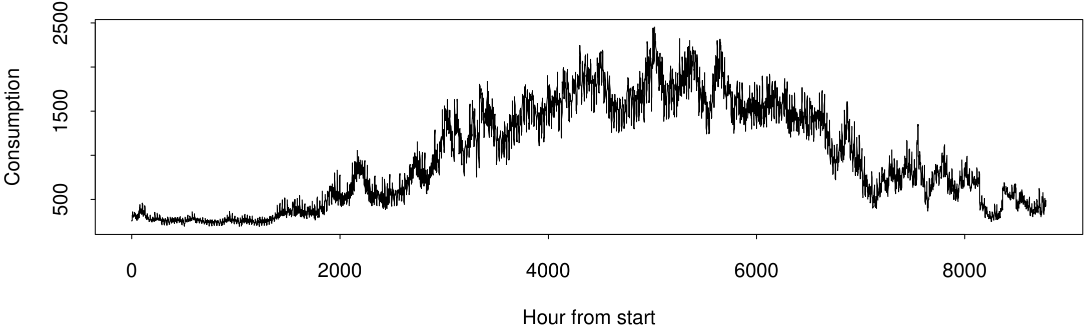
- For example: Try to keep $K_t$ constant at $S_0$, which leads to:

$$\lambda(t) \simeq 1 - \frac{\varepsilon_t^2}{S_0 \left[1 + \boldsymbol{x}_t^T \boldsymbol{P}_{t-1} \boldsymbol{x}_t\right]}$$

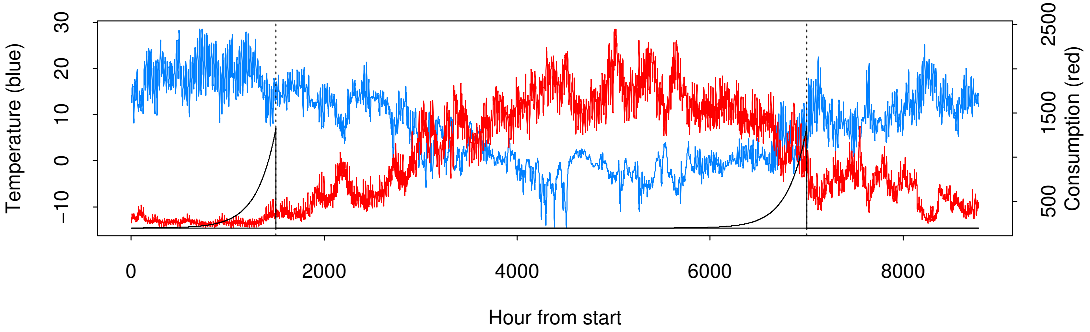
- What is the biggest danger in using variable forgetting? and how to solve it?
- Apply a lower bound $\lambda_{\text{min}}$ on $\lambda(t)$
- Finally, try different values of $S_0$ to find one that results in an acceptable rate of updating parameter estimates.

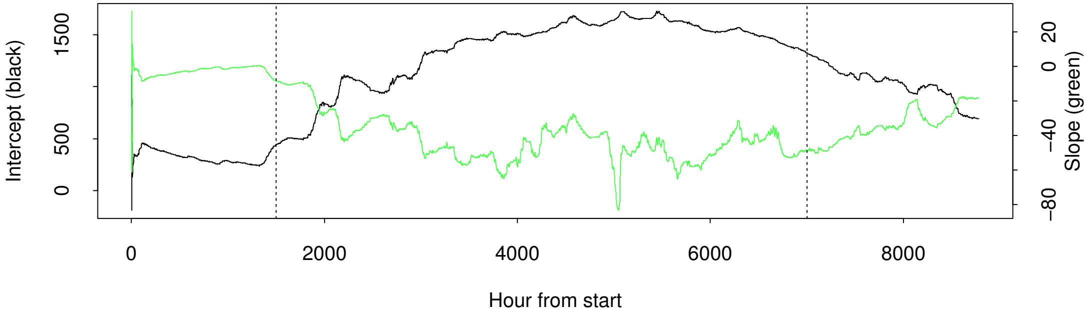
# Example: $HC_t = \mu + \theta_1 T_t + \varepsilon_t$



# Example: $HC_t = \mu + \theta_1 T_t + \varepsilon_t$, $\lambda = 0.995$



# Recursive pseudo-linear regression

- Problem: The ARMA structure cannot be estimated using regression directly. why?
- How to circumvent this problem?
- Given the parameters, $\boldsymbol{\theta}$, the one-step prediction and thereby residuals can be calculated and used for regression.
- The model becomes

$$\widehat{Y}_{t|t-1}(\boldsymbol{\theta}) = \boldsymbol{X}_t^T(\boldsymbol{\theta})\boldsymbol{\theta},$$

  where $\boldsymbol{X}_t^T(\boldsymbol{\theta})$ contains columns of residuals, assuming that the first few ones are zero.
- We minimize

$$S_t(\boldsymbol{\theta}) = \lambda(t)S_{t-1}(\boldsymbol{\theta}) + (Y_t - \boldsymbol{X}_t^T(\boldsymbol{\theta})\theta)^2$$

  with respect to $\boldsymbol{\theta}$.
- Then, the RPLR algorithm is:

$$\begin{aligned} \boldsymbol{R}_t &= \boldsymbol{x}_t\boldsymbol{x}_t^T + \lambda(t)\boldsymbol{R}_{t-1} \\ \boldsymbol{h}_t &= \boldsymbol{x}_t Y_t + \lambda(t)\boldsymbol{h}_{t-1} \\ \widehat{\boldsymbol{\theta}}_t &= \boldsymbol{R}_t^{-1}\boldsymbol{h}_t \end{aligned}$$

- I.e. as before except that $\boldsymbol{x}_t$ must be calculated at each step.

# Model-based adaptive estimation

- What is this?

$$\begin{aligned} \boldsymbol{X}_{t+1} &= \boldsymbol{X}_t + \boldsymbol{e}_{1,t}, \quad V(\boldsymbol{e}_{1,t}) = \boldsymbol{\Sigma}_1 \\ Y_t &= \boldsymbol{C}_t \boldsymbol{X}_t + \boldsymbol{e}_{2,t}, \quad V(\boldsymbol{e}_{2,t}) = \boldsymbol{\Sigma}_2 \end{aligned}$$

- How do we predict and reconstruct such a system?
- As shown in the previous lecture we can include the parameters as the latent state(s)

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \boldsymbol{e}_{1,t}, \quad V(\boldsymbol{e}_{1,t}) = \boldsymbol{\Sigma}_1 \\ Y_t &= \boldsymbol{X}_t^T \boldsymbol{\theta}_t + \boldsymbol{e}_{2,t}, \quad V(\boldsymbol{e}_{2,t}) = \boldsymbol{\Sigma}_2 \end{aligned}$$

- The parameters might even follow a linear model:

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{A}\boldsymbol{\theta}_t + \boldsymbol{e}_{1,t}, \quad V(\boldsymbol{e}_{1,t}) = \boldsymbol{\Sigma}_1 \\ Y_t &= \boldsymbol{X}_t^T \boldsymbol{\theta}_t + \boldsymbol{e}_{2,t}, \quad V(\boldsymbol{e}_{2,t}) = \boldsymbol{\Sigma}_2 \end{aligned}$$