Time Series Analysis

Lasse Engbo Christiansen

Department of Applied Mathematics and Computer Science
Technical University of Denmark

December 8, 2017

Outline of the lecture

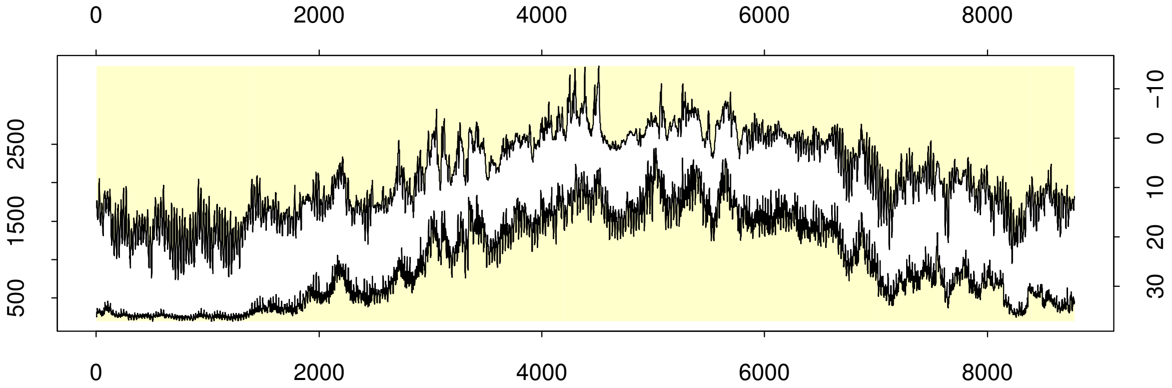
Recursive and adaptive estimation:

- Introduction to Chapter 11
- Recursive LS, Section 11.1
- Recursive pseudo-linear regression, Section 11.2
- Model-based adaptive estimation, Section 11.4

Further topics:

- Non-linear Time Series
- Stochastic differential equations
- Cointegration

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:

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

FIR:

$$\begin{aligned} Y_t &= \mu + \omega(B) U_t + \varepsilon_t \\ &= \mu + \omega_0 U_t + \omega_1 U_{t-1} + + \ldots \omega_s U_{t-s} + \varepsilon_t \end{aligned}$$

AR:

$$\begin{aligned} \phi(B) Y_t &= \mu + \varepsilon_t \Leftrightarrow \\ Y_t &= \mu - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \ldots - \phi_p Y_{t-p} + \varepsilon_t \end{aligned}$$

ARX:

$$\begin{aligned} \phi(B) Y_t &= \mu + \omega(B) U_t + \varepsilon_t \Leftrightarrow \\ Y_t &= \mu - \phi_1 Y_{t-1} - \ldots - \phi_p Y_{t-p} + \omega_0 U_t + \ldots + \omega_s U_{t-s} + \varepsilon_t \end{aligned}$$

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 (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:

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

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

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:

- $\boldsymbol{R}_0 = \boldsymbol{0}$ (matrix of zeros)
- $\boldsymbol{h}_0 = \boldsymbol{0}$ (vector of zeros)
- Wait with $\widehat{\boldsymbol{\theta}}_t$ until $\boldsymbol{R}_t$ is invertible

The RLS algorithm – 2 equivalent formulations

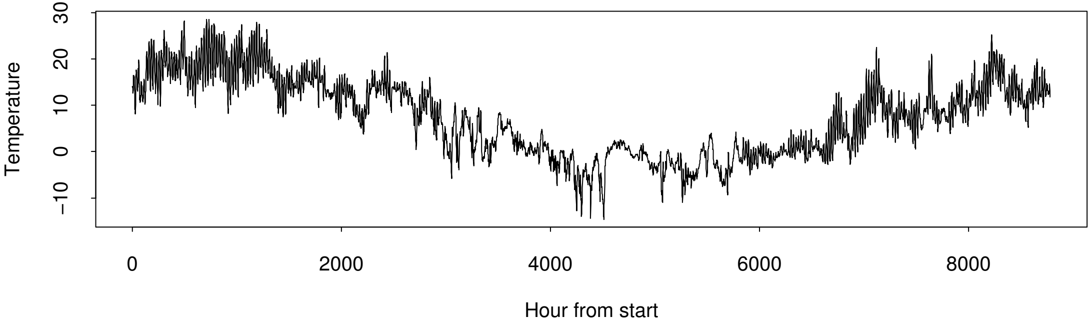
1. Eliminating $\boldsymbol{h}_t$:

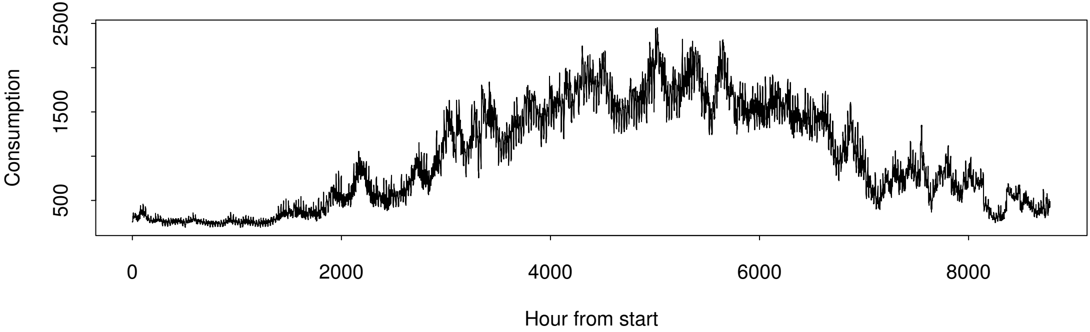
$$\begin{aligned} \boldsymbol{R}_t &= \boldsymbol{x}_t \boldsymbol{x}_t^T + \boldsymbol{R}_{t-1} \\ \widehat{\boldsymbol{\theta}}_t &= \widehat{\boldsymbol{\theta}}_{t-1} + \boldsymbol{R}_t^{-1} \boldsymbol{x}_t (Y_t - \boldsymbol{x}_t^T \widehat{\boldsymbol{\theta}}_{t-1}) \end{aligned}$$

2. Eliminating $\boldsymbol{h}_t$ and avoiding matrix-inversion $P = R^{-1}$:

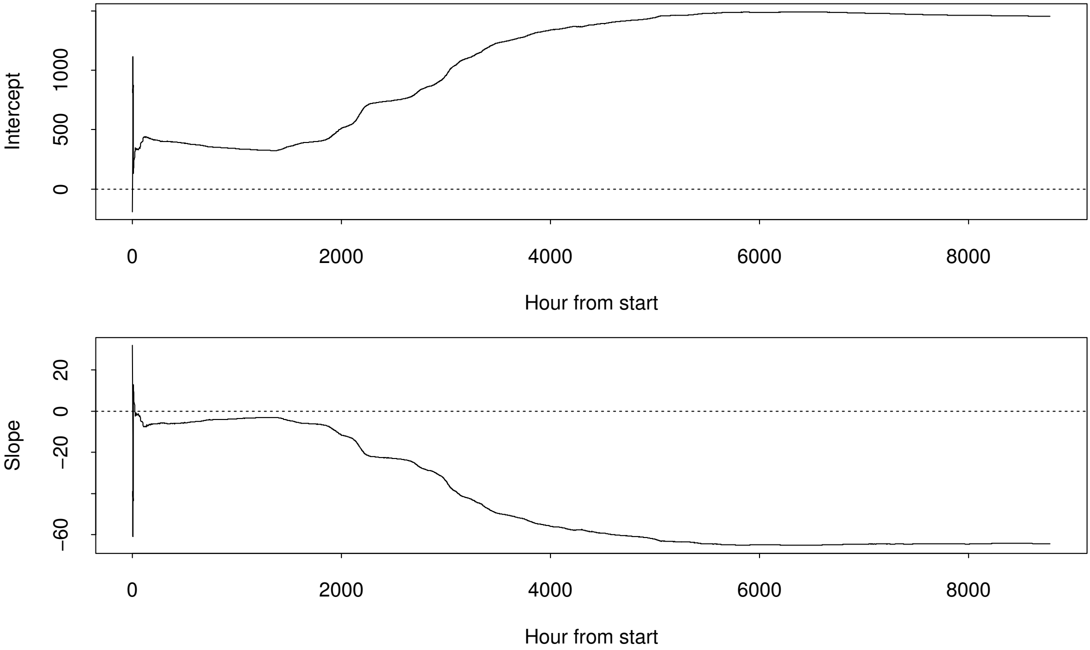
$$\begin{aligned} \boldsymbol{K}_t &= \frac{\boldsymbol{P}_{t-1} \boldsymbol{x}_t}{1 + \boldsymbol{x}_t^T \boldsymbol{P}_{t-1} \boldsymbol{x}_t} \\ \widehat{\boldsymbol{\theta}}_t &= \widehat{\boldsymbol{\theta}}_{t-1} + \boldsymbol{K}_t (Y_t - \boldsymbol{x}_t^T \widehat{\boldsymbol{\theta}}_{t-1})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

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
- If we want a method which forgets old observations we apply weights which start at 1 and goes to 0 when observations gets old:

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

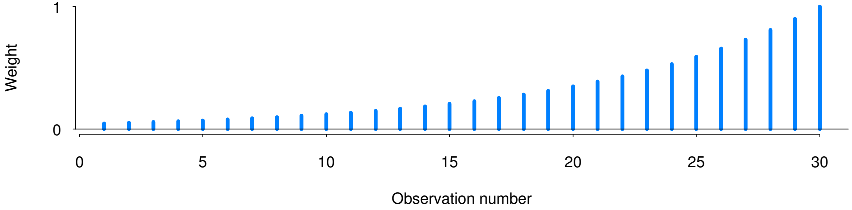
$$S_t(\boldsymbol{\theta}) = \sum_{s=1}^{t} \beta(t,s)(Y_s - \boldsymbol{x}_s^T \boldsymbol{\theta})^2$$

  where $\boldsymbol{W} = \text{diag}(\beta(t,1), \beta(t,2), \dots, \beta(t,t-1), 1)$
- $\beta(t,s)$ express how we assign weights to old observations

Exponential decay of weights

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

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



- 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

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 $\boldsymbol{h}_t$:

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

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 λ 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 λ can be found by "trial and error"
- It is often a good idea to select the values of λ to be investigated so that the corresponding values of T_0 are approximately equidistant
- The criteria to evaluate may depend on the application, but the sum of squared one-step prediction errors is often appropriate
- An initialization period should be excluded from the evaluation

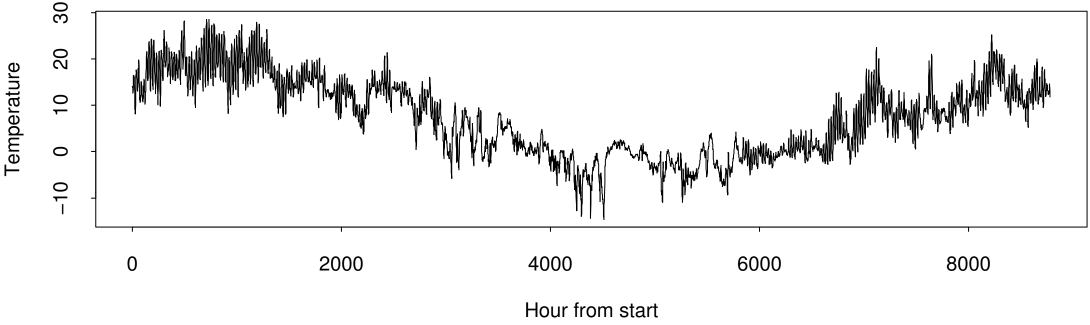
Variable forgetting

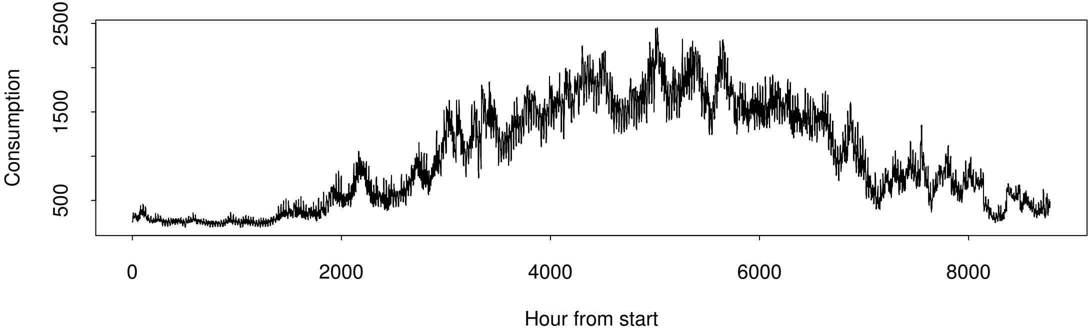
- Many methods exists
- One is based on the aim of keeping the criteria functions defining the estimates constant at S_0
- Leads to:

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

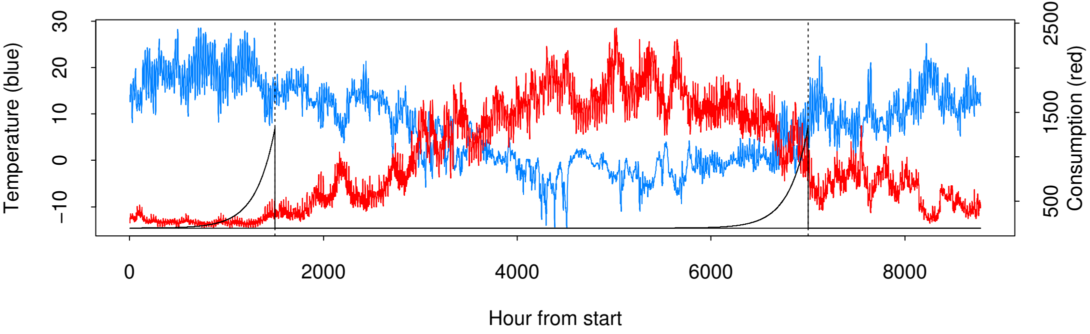
- A lower bound λ_{min} on $\lambda(t)$ should be applied
- For optimal tuning of this method S_0 could be varied

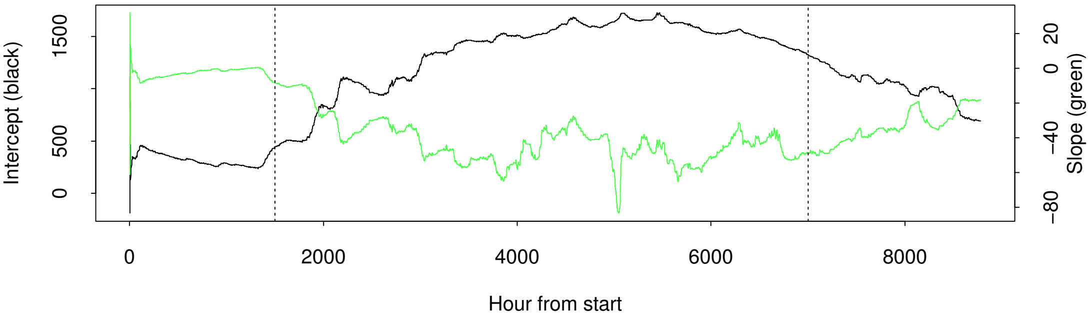
Example: $HC_t = \mu + \theta_1 T_t + \varepsilon_t$





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.
- However, given the parameters, $\boldsymbol{\theta}$, the one-step prediction residuals can be calculated and used for regression.
- The model becomes

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

- 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?
- Now the parameters are the latent state

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

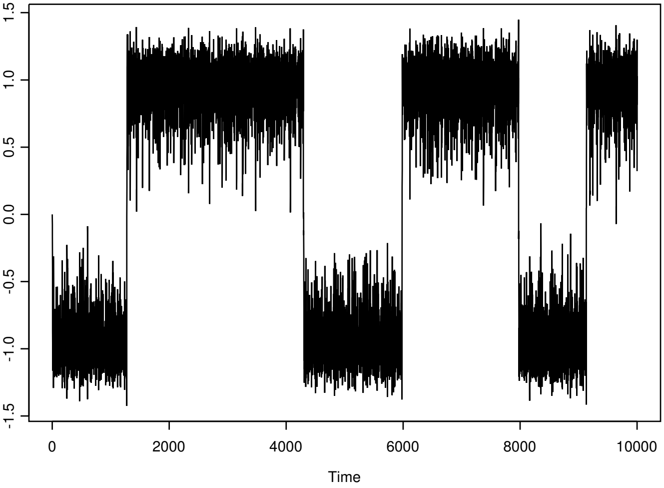
- That means we can use the Kalman filter for tracking the parameters.
- See the formulation of the Kalman filter in the book.

Models with time-varying parameters

- ARMA processes where the parameters are time-varying.
- The parameters can follow deterministic functions of time or be stochastic processes.
- When the parameters are stochastic processes, the models are called *double stochastic*.
  - This could be an ARMA structure where the parameters are other ARMA processes.
- The Kalman filter is the central tool in estimation on such processes.

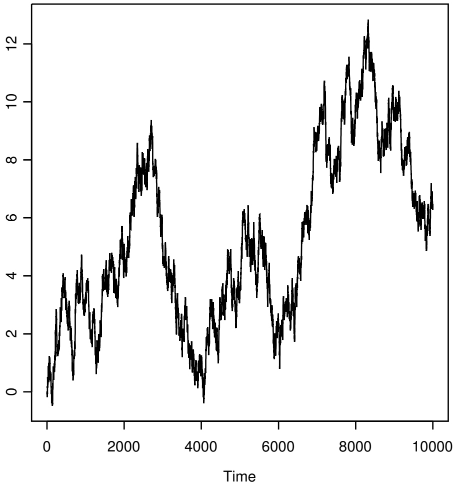
Non-linear time series

There are of course many, many possible formulations.



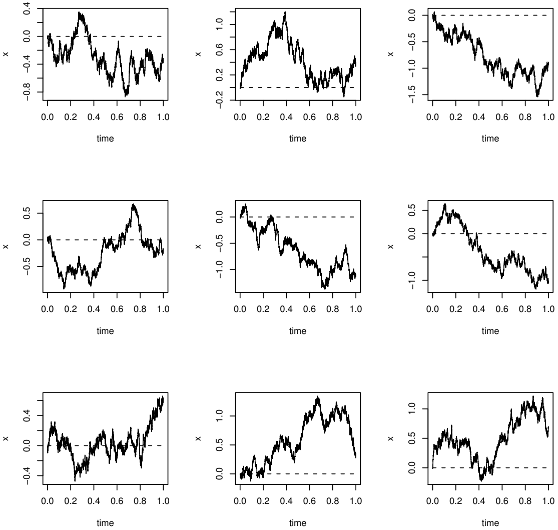
$dX_t = -X_t(X_t + 1)(X_t - 1)dt + \sigma dW_t$,
where $\sigma = 0.11$, and W is a Brownian Motion.

Brownian Motion



The continuous-time counterpart to a random walk

9 Brownian Motions



Properties of a Brownian Motion W

- The stochastic process W is continuous with probability 1;
- The stochastic process W is nowhere differentiable with probability 1;
- If you wish to straighten out the graph over any interval, the length will be infinite due to infinite variation.

Stochastic Differential Equations

The equation

$$dX_t = -X_t(X_t+1)(X_t-1)dt + \sigma dW_t$$

is a Stochastic Differential Equation (SDE).
More general SDE's:

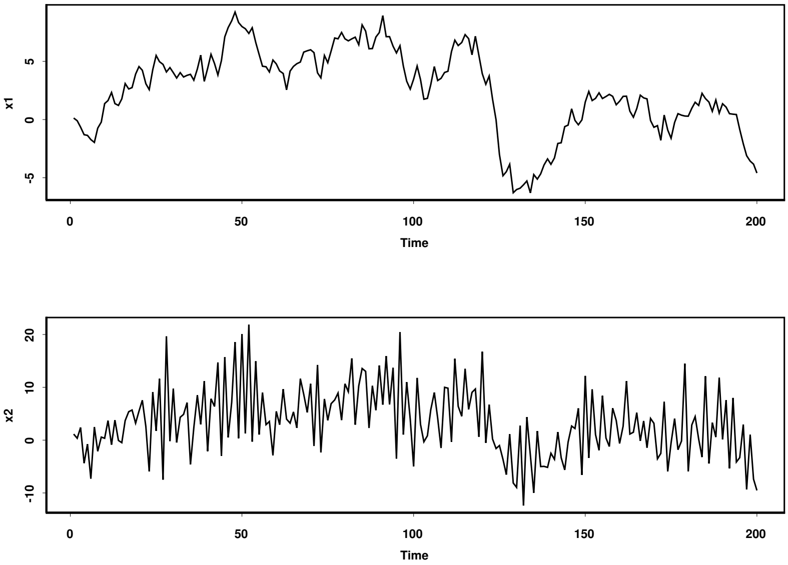
$$dX_t = a(X,t)dt + b(X,t)dW_t$$

The equation is really an integral equation:

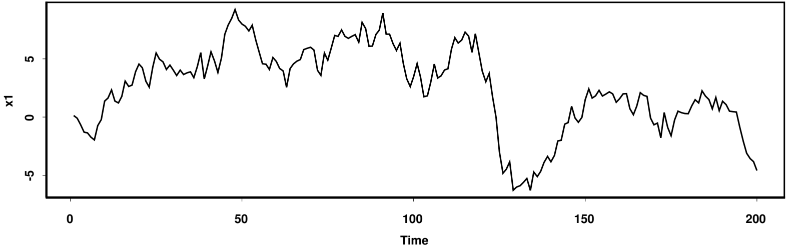
$$X(t) = \int_0^t a(X,s)\,ds + \int_0^t b(X,s)\,dW(s)$$

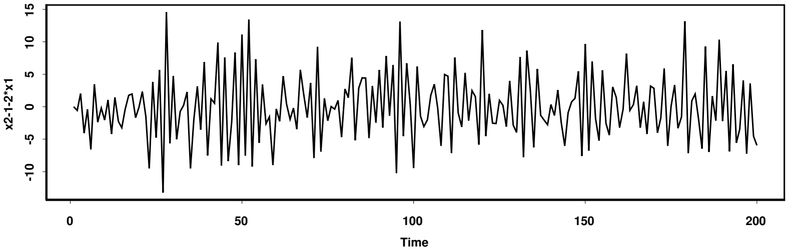
The last term is a so-called *stochastic integral* (advanced topic).

Cointegration 1



Cointegration 2





Same Time Series in different coordinates

Cointegration links

- Cointegration in economics:
  http://isc.temple.edu/economics/notes/cointegration/cointegration.HTM
- Clive Grangers lecture:
  http://www.nobelprize.org/nobel_prizes/economics/laureates/2003/granger-lecture.pdf
- Justification from the Swedish Academy of Sciences (much like link 1):
  http://www.nobelprize.org/nobel_prizes/economics/laureates/2003/advanced-economicsciences2003.pdf

More links:

- Spatial time series (Clorophyll dynamics): http://spg.ucsd.edu/Satellite_Projects/Chlorophyll_dynamics_Santa_Barbara_Basin/Chlorophyll_dynamics_Santa_Barbara_Basin.htm
- Sunspot research at NASA: http://science.nasa.gov/science-news/science-at-nasa/2008/11jul_solarcycleupdate/
- Time Series Data Library by Rob J Hyndman: http://datamarket.com/data/list/?q=provider:tsdl

Related courses

- 02427 Advanced time series analysis
- 02425 Diffusions and stochastic differential equations
- 02407 Stochastic processes
- Many more ;-)

Highlights

- Recursive LS for many types of models.
- Adaptive Recursive LS
- Kalman filtering to trace parameters
- Where next:
  - Continuous time
  - Non-linearities
  - Stochastic Differential Equations

Merry Christmas!