Forecasting & Predictive Analytics

Guillaume Chevillon (chevillon@essec.edu)
and Pierre Jacob (jacob@essec.edu)

October-December 2021
4th set of slides
State space models

Overview

- The class of ARMA models is central for time series analysis and forecasting, and can be used to approximate many stationary processes. ARIMA and SARIMA extensions can account for stochastic trends and seasonality.
- These are instances of state space models (SSM), a useful, encompassing formulation that also include many other models.
- In SSMs, the observed series is viewed as noisy measurements of a latent, underlying process, modeled as a Markov chain.
- Key questions with SSMs are filtering, smoothing and likelihood evaluation, which can be done with Kalman filters.

Markov chains

Definition

A stochastic process (X_t) is a Markov chain if X_t given X_{t-1} is independent of $X_0, \dots, X_{t-2}$.

Focus here on discrete time processes, there are also continuous-time Markov chains.

We have seen some examples already:

- random walk processes,
- AR(1) processes. . . what about AR(p)?

Definition

A process is Markov of order m if X_t given $X_{t-1}, \ldots, X_{t-m}$ is independent of $X_0, \ldots, X_{t-m-1}$.

A Markov chain (X_t) is homogeneous if the distribution of X_t given X_{t-1} is the same as the distribution of X_1 given X_0.

(Think about Dory in *Finding Nemo*.)

More examples

- A person goes on a webpage, and clicks on one of the links at random, ends up on a new webpage and starts again.

  This underpins “PageRank”, see *Markov Chains & PageRank* by Roger Wattenhofer.

- Consider the number of persons queuing at the cafetaria. . .

- Consider a population of organisms (animals, trees), that either produce offsprings at each time step, or disappear.

Animal movements with Markov models

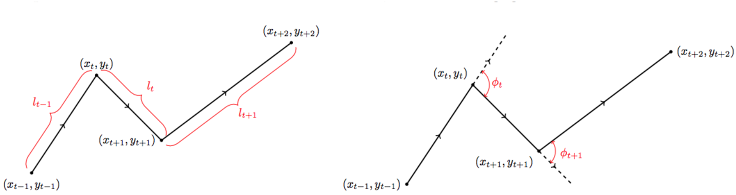


Figure 1: *Illustration of step lengths and turning angles*

Taken from the vignette of the package `moveHMM`.

https://cran.r-project.org/web/packages/moveHMM/

Definition on discrete spaces

Let (X_t) be a Markov chain taking values in a finite space $\mathbb{X} = \{1, 2, \ldots, K\}$.

Assume $X_0 \sim \pi_0$, where $\pi_0 = (\pi_0(1), \ldots, \pi_0(K))$, is a vector of non-negative real values summing to one.

Let $P_{ij} = \mathbb{P}(X_t = j | X_{t-1} = i)$, for $i, j \in \mathbb{X}$. The matrix P is $K \times K$, and its rows sum to one.

The Markov chain is entirely specified by π_0 and P.

Interactive visualization

http://setosa.io/ev/markov-chains/
by Victor Powell

Joint probability

Using the Markov property,

$$\begin{aligned}
&\mathbb{P}(X_0 = x_0, \dots, X_t = x_t) \\
&= \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}, \dots, X_0 = x_0)\mathbb{P}(X_0 = x_0, \dots, X_{t-1} = x_{t-1}) \\
&= \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1})\mathbb{P}(X_0 = x_0, \dots, X_{t-1} = x_{t-1}) \\
&= \left\{ \prod_{s=1}^{t} \mathbb{P}(X_s = x_s | X_{s-1} = x_{s-1}) \right\} \mathbb{P}(X_0 = x_0) \\
&= \left\{ \prod_{s=1}^{t} P_{x_{s-1} x_s} \right\} \pi_0(x_0).
\end{aligned}$$

Continuous spaces: same thing

Let (X_t) be a Markov chain taking values in a continuous space $\mathbb{X} = \mathbb{R}^d$, where d indicates the dimension.

We write π_0 for the initial distribution of X_0.

We write $P(x_t|x_{t-1})$ for the density of the transition probability from $X_{t-1} = x_{t-1}$, evaluated at x_t.

The joint density $p(x_0, \dots, x_t)$ can be written

$$p(x_0, \dots, x_t) = \pi_0(x_0) \prod_{s=1}^{t} P(x_s|x_{s-1}).$$

We find an explicit formula e.g. in the case of AR(1) processes.

Inference for Markov models

Markov chains can be proposed as models for time series data, e.g. autoregressive processes.

Can easily be extended to model nonlinear phenomena, e.g.

$$X_t = f(X_{t-1}) + W_t,$$

where f might not be linear.

Example: threshold AR(1) process (Tong 1978, 1983),

$$X_t = \rho X_{t-1} + \gamma \mathbb{1}(X_{t-2} > \delta) X_{t-1} + W_t.$$

When $\rho = 1$ and $\gamma < 0$ think of exchange rates: the Central Bank adopts a laissez-faire approach (the random walk) unless the exchange rate diverges too much from the target.

The joint density $p(x_0, \ldots, x_t)$ can be seen as a likelihood function, and optimized over the parameters (e.g. ρ, γ, δ above).

State space models

State space models

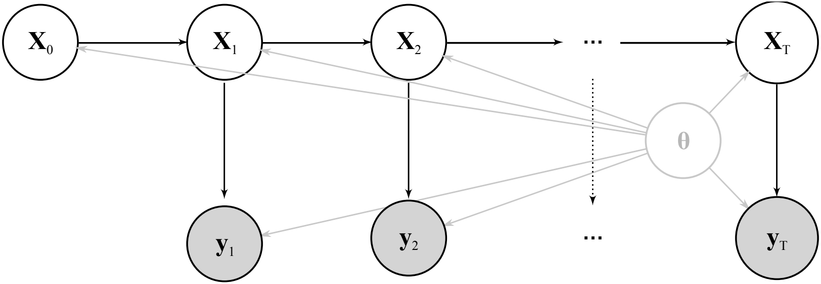


Diagram representation of the variables in a state space model. Also called a hidden Markov model.

Example: stochastic volatility model

Denote the daily price of an asset by p_t, at time t.

Define the observations to be the log-returns: $Y_t = \log(p_t / p_{t-1})$.

A (simple) stochastic volatility model assumes

$$\begin{aligned} Y_t &\sim \mathcal{N}(0, \exp(X_t)) \\ X_t &= \varphi X_{t-1} + W_t. \end{aligned}$$

More complicated models can be put on (X_t), e.g. with jumps.

e.g. book of Tsay, *Analysis of financial time series*, 2005.

Example: stochastic volatility model

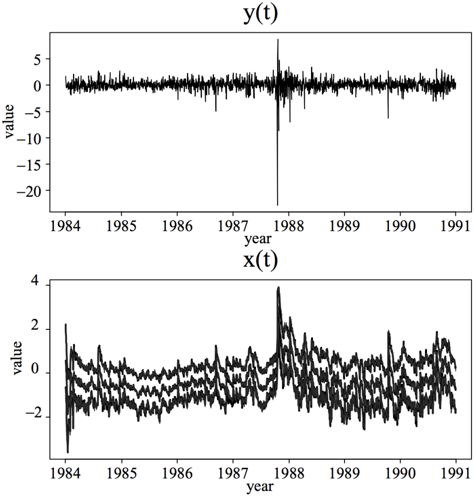


Figure: Log-returns of a financial asset (top) and estimated stochastic volatility (bottom) over time.

Linear Gaussian state space models

Observations (Y_t). The process (X_t) is “hidden” or “latent”.

$$
\begin{aligned}
Y_t &= AX_t + V_t, \text{ with } V_t \sim \mathcal{N}(0, \Sigma_V), \\
X_t &= \Phi X_{t-1} + W_t \text{ with } W_t \sim \mathcal{N}(0, \Sigma_W).
\end{aligned}
$$

Parameters: A, Φ, Σ_V, Σ_W. We also need to specify X_0, e.g. $\mathcal{N}(m_0, C_0)$.

ARMA as state space models

Define $r = \max(p, q + 1)$, and extend φ or θ with zeros. Consider a latent process X_t made of r elements $(X_{t,1}, \dots, X_{t,r})$ such that

$$\begin{pmatrix} X_{t,1} \\ X_{t,2} \\ \vdots \\ X_{t,r-1} \\ X_{t,r} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ \varphi_r & \varphi_{r-1} & \varphi_{r-2} & \dots & \varphi_1 \end{pmatrix} \begin{pmatrix} X_{t-1,1} \\ X_{t-1,2} \\ \vdots \\ X_{t-1,r-1} \\ X_{t-1,r} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} W_t.$$

Define the observation equation as

$$Y_t = \begin{pmatrix} \theta_{r-1} & \theta_{r-2} & \dots & \theta_1 & 1 \end{pmatrix} X_t + V_t,$$

where V_t is zero for all times t.

Structural models

- Local level model or “random walk plus noise”:

$$Y_t = X_t + V_t, \text{ with } V_t \sim \mathcal{N}(0, \sigma_V^2),$$
$$X_t = X_{t-1} + W_t \text{ with } W_t \sim \mathcal{N}(0, \sigma_W^2).$$

- Local linear trend model:

$$\begin{pmatrix} X_t \\ B_t \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X_{t-1} \\ B_{t-1} \end{pmatrix} + \begin{pmatrix} W_t \\ W_t^b \end{pmatrix},$$
$$Y_t = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} X_t \\ B_t \end{pmatrix} + V_t.$$

Structural models

We can add a seasonal component S_t, with period s. Observation equation:

$$Y_t = \begin{pmatrix} 1 & 0 & 1 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} X_t \\ B_t \\ S_t \\ S_{t-1} \\ \vdots \\ S_{t-s+1} \end{pmatrix} + V_t.$$

The state equation:

$$\begin{pmatrix} X_t \\ B_t \\ S_t \\ S_{t-1} \\ \vdots \\ S_{t-s+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & -1 & -1 & \dots & -1 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{pmatrix} \begin{pmatrix} X_{t-1} \\ B_{t-1} \\ S_{t-1} \\ S_{t-2} \\ \vdots \\ S_{t-s} \end{pmatrix} + \begin{pmatrix} W_t \\ W_t^b \\ W_t^s \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Filtering, smoothing, prediction and likelihood

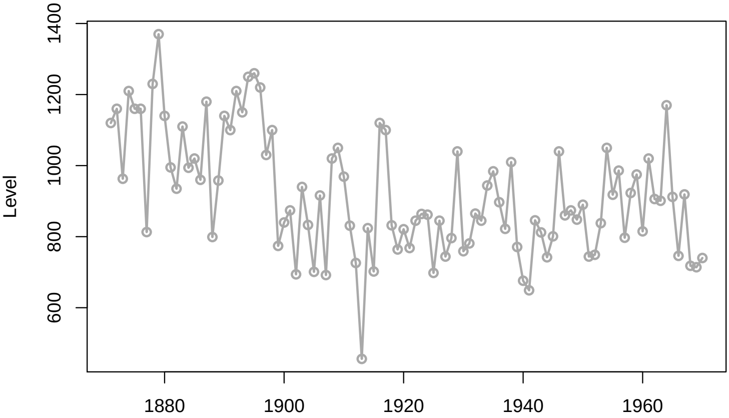
- Filtering: “tracking” X_t given $y_1, \ldots, y_t$.
- Smoothing: “recovering” X_t, given $y_1, \ldots, y_n$, with $t \leq n$.
- Prediction: “guessing” X_t, Y_t given $y_1, \ldots, y_n$, with $t > n$.
- The likelihood $p(y_1, \ldots, y_n|\beta)$ can be computed recursively

  $$\log p(y_1, \ldots, y_n|\beta) = \log p(y_1|\beta) + \sum_{t=2}^{n} \log p(y_t|y_1, \ldots, y_{t-1}, \beta),$$

  using filtering results.

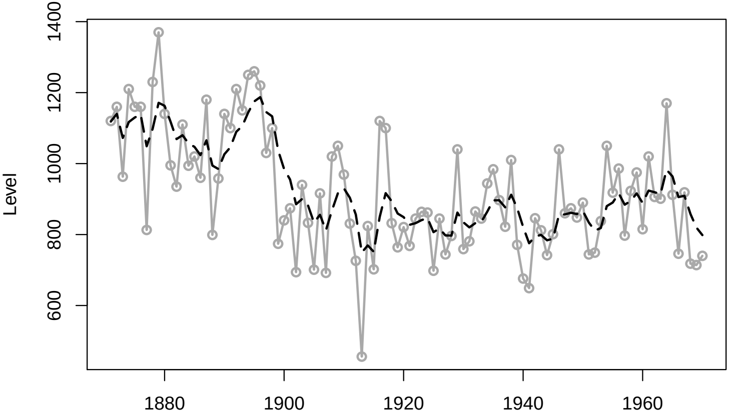
Nile river annual flow at Aswan, 1871–1970

Observed series.



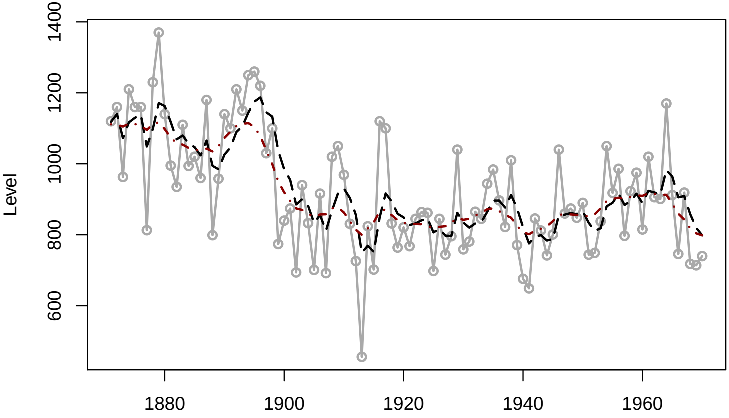
Nile river annual flow at Aswan, 1871–1970

Filtering with a local level model.



Nile river annual flow at Aswan, 1871–1970

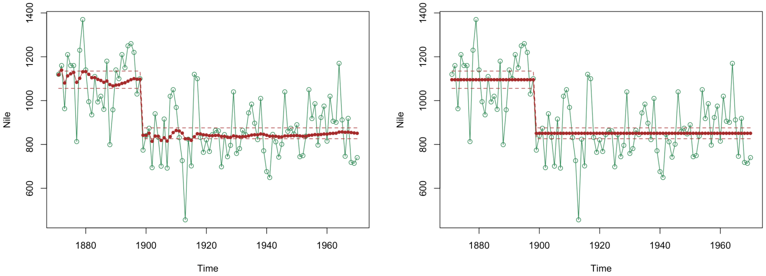
Smoothing with a local level model.



Nile river annual flow at Aswan, 1871–1970

Related model with stochastic volatility: innovations to latent space have zero variance except in 1899.

Filtered and smoothed estimates



Kalman filter

Goal

- The Kalman filter refers to an algorithm to perform filtering in linear Gaussian models.
  That is, it computes the mean and variance of X_t given $y_1, \ldots, y_t$, recursively.

- As a by-product it provides evaluations of the likelihood in $\mathcal{O}(n)$ operations.

- The Kalman smoother will be introduced later, to perform smoothing, i.e. computing the mean and variance of X_t given $y_1, \ldots, y_n$, for $1 \leq t \leq n$.

Initialization

Introduce $m_{t|s}$ and $C_{t|s}$, the conditional mean and variance

$$\begin{aligned} m_{t|s} &= \mathbb{E}[X_t | Y_1, \dots, Y_s], \\ C_{t|s} &= \mathbb{V}[X_t | Y_1, \dots, Y_s], \end{aligned}$$

for times t, s.

Start from known values for $m_{0|0}$ and $C_{0|0}$ such that

$$X_0 \sim \mathcal{N}(m_{0|0}, C_{0|0}).$$

Multivariate Normal cheatsheet

Multivariate Normal X, vector λ:

$$X \sim \mathcal{N}(\mu, \Sigma) \quad \Leftrightarrow \quad \lambda^T X \sim \mathcal{N}\left(\lambda^T \mu, \lambda^T \Sigma \lambda\right),$$

i.e. $X_1 + X_2 \sim \mathcal{N}(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$ if X_1, X_2 independent.
With “blocks” X_1 and X_2:

$$X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{21} & \Sigma_2 \end{pmatrix}\right).$$

Conditional distribution:

$$X_1 | X_2 = x_2 \sim \mathcal{N}\left(\mu_1 + \Sigma_{12}\Sigma_2^{-1}(x_2 - \mu_2), \Sigma_1 - \Sigma_{12}\Sigma_2^{-1}\Sigma_{21}\right).$$

Linear transform of X of size n, with a rectangular matrix A of size $p \times n$ and a vector b of size p:

$$AX + b \sim \mathcal{N}\left(A\mu + b, A\Sigma A^T\right).$$

Linear Gaussian models

We consider the model:

$$\begin{aligned} Y_t &= AX_t + V_t, \text{ with } V_t \sim \mathcal{N}(0, \Sigma_V), \\ X_t &= \Phi X_{t-1} + W_t, \text{ with } W_t \sim \mathcal{N}(0, \Sigma_W). \end{aligned}$$

Parameters: A, Φ, Σ_V, Σ_W.

Start from known values for $m_{0|0}$ and $C_{0|0}$.

Step zero to step one (prediction)

We are told $X_0 \sim \mathcal{N}(m_{0|0}, C_{0|0})$: we know where X_0 is *a priori*.

$X_1 = \Phi X_0 + W_1$, equivalently $X_1 \sim \mathcal{N}(\Phi m_{0|0}, \Phi C_{0|0} \Phi^T + \Sigma_W)$.

Therefore $m_{1|0} = \Phi m_{0|0}$ and $C_{1|0} = \Phi C_{0|0} \Phi^T + \Sigma_W$.

Still a priori, $X_1 \sim \mathcal{N}(m_{1|0}, C_{1|0})$.

Then we observe y_1, and recall that $Y_1 = AX_1 + V_1$.

Step one (update)

Joint Normal distribution ($Y_1 = AX_1 + V_1$)

$$\begin{pmatrix} X_1 \\ Y_1 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m_{1|0} \\ Am_{1|0} \end{pmatrix}, \begin{pmatrix} C_{1|0} & C_{1|0}A^T \\ AC_{1|0} & AC_{1|0}A^T + \Sigma_V \end{pmatrix} \right).$$

We apply the formula for the conditional Normal distribution:

$$\begin{aligned} m_{1|1} &= m_{1|0} + C_{1|0}A^T(AC_{1|0}A^T + \Sigma_V)^{-1}\left(y_1 - Am_{1|0}\right), \\ C_{1|1} &= C_{1|0} - C_{1|0}A^T(AC_{1|0}A^T + \Sigma_V)^{-1}AC_{1|0}. \end{aligned}$$

A posteriori, $X_1|Y_1 \sim \mathcal{N}(m_{1|1}, C_{1|1})$.

Step one to step two (prediction)

We now live in the *conditional world* where $Y_1 = y_1$.

We are told: $X_2 = \Phi X_1 + W_2$,
and $X_1 \sim \mathcal{N}(m_{1|1}, C_{1|1})$ (in the conditional world).

We obtain $X_2 \sim \mathcal{N}(\Phi m_{1|1}, \Phi C_{1|1} \Phi^T + \Sigma_W)$.

We write

$$\begin{aligned} m_{2|1} &= \Phi m_{1|1}, \\ C_{2|1} &= \Phi C_{1|1} \Phi^T + \Sigma_W. \end{aligned}$$

Then we observe y_2, and recall that $Y_2 = A X_2 + V_2$.

Step two (update)

We live in the *conditional world* where $Y_1 = y_1$.

Joint Normal distribution ($Y_2 = AX_2 + V_2$)

$$\begin{pmatrix} X_2 \\ Y_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m_{2|1} \\ m_{2|1} \end{pmatrix}, \begin{pmatrix} C_{2|1} & C_{2|1}A^T \\ AC_{2|1} & AC_{2|1}A^T + \Sigma_V \end{pmatrix} \right).$$

We apply the formula for the conditional Normal distribution:

$$\begin{aligned} m_{2|2} &= m_{2|1} + C_{2|1}A^T(AC_{2|1}A^T + \Sigma_V)^{-1}\left(y_2 - Am_{2|1}\right), \\ C_{2|2} &= C_{2|1} - C_{2|1}A^T(AC_{2|1}A^T + \Sigma_V)^{-1}AC_{2|1}. \end{aligned}$$

A posteriori, $X_2|Y_2 \sim \mathcal{N}(m_{2|2}, C_{2|2})$. And so on!

Kalman filter for linear Gaussian models

Inputs: $m_{0|0}$, $C_{0|0}$, A, Φ, Σ_V, Σ_W, $y_1, y_2, \ldots, y_n$.
Ouput: $m_{t|t}$, $C_{t|t}$ for all $t = 1, \ldots, n$.
Prediction (X_t given $y_1, \ldots, y_{t-1}$):

$$\begin{aligned} m_{t|t-1} &= \Phi m_{t-1|t-1}, \\ C_{t|t-1} &= \Phi C_{t-1|t-1} \Phi^T + \Sigma_W. \end{aligned}$$

Update (X_t given $y_1, \ldots, y_t$):

$$\begin{aligned} m_{t|t} &= m_{t|t-1} + C_{t|t-1} A^T (A C_{t|t-1} A^T + \Sigma_V)^{-1} \left(y_t - A m_{t|t-1} \right), \\ C_{t|t} &= C_{t|t-1} - C_{t|t-1} A^T (A C_{t|t-1} A^T + \Sigma_V)^{-1} A C_{t|t-1}. \end{aligned}$$

Kalman filter for linear Gaussian models

It is common to introduce the Kalman gain:

$$K_t = C_{t|t-1} A^T \left(A C_{t|t-1} A^T + \Sigma_V \right)^{-1}.$$

Then the update reads

$$\begin{aligned} m_{t|t} &= m_{t|t-1} + K_t \left(y_t - A m_{t|t-1} \right), \\ C_{t|t} &= \left(I - K_t A \right) C_{t|t-1}. \end{aligned}$$

Likelihood evaluation

In the conditional world given $y_1, \ldots, y_{t-1}$,

$$X_t \sim \mathcal{N}(m_{t|t-1}, C_{t|t-1}),$$

and thus

$$Y_t = AX_t + V_t \sim \mathcal{N}(Am_{t|t-1}, AC_{t|t-1}A^T + \Sigma_V).$$

We deduce:

$$p(y_t|y_{1:t-1}, \beta) = \mathcal{N}(y_t; Am_{t|t-1}, AC_{t|t-1}A^T + \Sigma_V).$$

$\Rightarrow$ We can compute $p(y_t|y_{1:t-1}, \beta)$ after the prediction step.

More general models

We can make A and Φ vary with t, and add inputs:

$$\begin{aligned} Y_t &= A_t X_t + \Gamma U_t + V_t, \text{ with } V_t \sim \mathcal{N}(0, \Sigma_V), \\ X_t &= \Phi_t X_{t-1} + \Lambda U_t + W_t \text{ with } W_t \sim \mathcal{N}(0, \Sigma_W). \end{aligned}$$

Inputs allow to e.g. cover regression with autocorrelated errors, e.g.

$$Y_t = \Gamma U_t + \varepsilon_t,$$

where ε_t is an ARMA(p,q).

More general steps

If we start from $X_t \sim \mathcal{N}(m_{t-1|t-1}, C_{t-1|t-1})$, and

$$X_t = \Phi_t X_{t-1} + \Lambda U_t + W_t \text{ with } W_t \sim \mathcal{N}(0, \Sigma_W).$$

Then

$$\begin{aligned} m_{t|t-1} &= \Phi_t m_{t-1|t-1} + \Lambda U_t, \\ C_{t|t-1} &= \Phi_t C_{t-1|t-1} {\Phi_t}^T + \Sigma_W. \end{aligned}$$

Then,

$$Y_t = A_t X_t + \Gamma U_t + V_t, \text{ with } V_t \sim \mathcal{N}(0, \Sigma_V),$$

so that, with

$$K_t = C_{t|t-1} {A_t}^T (A_t C_{t|t-1} {A_t}^T + \Sigma_V)^{-1},$$

the update reads

$$\begin{aligned} m_{t|t} &= m_{t|t-1} + K_t \left(y_t - (A_t m_{t|t-1} + \Gamma U_t) \right), \\ C_{t|t} &= (I - K_t A_t) C_{t|t-1}. \end{aligned}$$

Filtering + likelihood

- Kalman filter requires one forward pass over the data.
  Computational cost in $\mathcal{O}(n)$.
  [Difficulties arise in high-dimensional models.]

- Yields the filtering means and variances.

- Likelihood evaluations derived from one-step predictions.

- We can also construct m-step predictions.

Kalman smoother

Smoothing

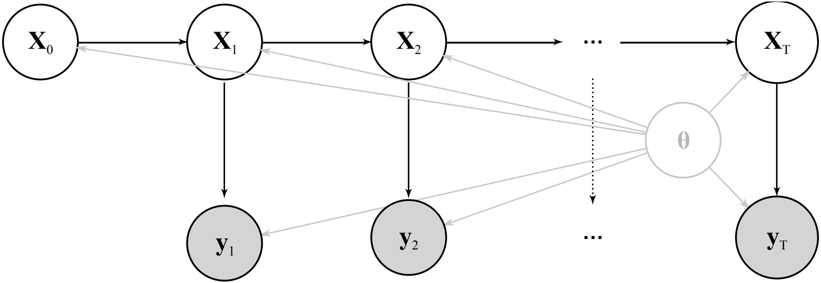
- Suppose that we have $m_{t|t}, C_{t|t}$ for all $t = 1, \dots, n$ (from the Kalman filter).

- We now want to find $m_{t|n}, C_{t|n}$ for all $t = 1, \dots, n$.

- Good news: we already have $m_{n|n}, C_{n|n}$, the *last ones*. We now envision a backward pass, $t = n - 1, \dots, 2, 1$.

- Let's see how to compute $m_{n-1|n}, C_{n-1|n}$.

Smoothing



Important remark:

$$
\begin{aligned}
p(x_{n-1}|x_n, y_{1:n}) &= p(x_{n-1}|x_n, y_{1:n-1}) \\
&= \frac{p(x_n|x_{n-1})p(x_{n-1}|y_{1:n-1})}{p(x_n|y_{1:n-1})},
\end{aligned}
$$

where y_n does not appear.

Smoothing

Given $y_1, \ldots, y_{n-1}$ (!important!),

$$\begin{pmatrix} X_{n-1} \\ X_n \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m_{n-1|n-1} \\ m_{n|n-1} \end{pmatrix}, \begin{pmatrix} C_{n-1|n-1} & C_{n-1|n-1}\Phi^T \\ \Phi C_{n-1|n-1} & C_{n|n-1} \end{pmatrix} \right),$$

thus $X_{n-1}|X_n = x_n$ follows a normal distribution with

$$\text{mean} = m_{n-1|n-1} + C_{n-1|n-1}\Phi^T C_{n|n-1}^{-1}(x_n - m_{n|n-1}),$$
$$\text{variance} = C_{n-1|n-1} - C_{n-1|n-1}\Phi^T C_{n|n-1}^{-1}\Phi C_{n-1|n-1}.$$

Smoothing

Introducing $L_{n-1} = C_{n-1|n-1} \Phi^T C_{n|n-1}^{-1}$,
$X_{n-1} | X_n = x_n$ follows a normal distribution with

$$\text{mean} = m_{n-1|n-1} + L_{n-1}(x_n - m_{n|n-1}),$$
$$\text{variance} = C_{n-1|n-1} - L_{n-1} C_{n|n-1} L_{n-1}^T.$$

This holds

- given $y_1, \ldots, y_{n-1}$, and also
- given all the observations, because of the model's structure.

Recall $X_n \sim \mathcal{N}(m_{n|n}, C_{n|n})$, given all the observations.

Smoothing

We retrieve the distribution of X_{n-1} using the tower property.

$$\begin{aligned}\mathbb{E}[X_{n-1}|y_{1:n}] &= \mathbb{E}[\mathbb{E}[X_{n-1}|X_n, y_{1:n}]|y_{1:n}] \text{ by tower property}\\ &= \mathbb{E}[\mathbb{E}[X_{n-1}|X_n, y_{1:n-1}]|y_{1:n}] \text{ by model's structure}\\ &= \mathbb{E}[m_{n-1|n-1} + L_{n-1}(X_n - m_{n|n-1})|y_{1:n}]\\ &= m_{n-1|n-1} + L_{n-1}(m_{n|n} - m_{n|n-1}).\end{aligned}$$

For the variance, we use

$$\begin{aligned}\mathbb{V}[X_{n-1}|y_{1:n}] = &\ \mathbb{E}[\mathbb{V}[X_{n-1}|X_n, y_{1:n-1}]|y_{1:n}]\\ &+ \mathbb{V}[\mathbb{E}[X_{n-1}|X_n, y_{1:n-1}]|y_{1:n}],\end{aligned}$$

and compute the two terms separately.

Note: we can write $\mathbb{E}[\cdot|y_{1:t}] = \mathbb{E}[\cdot|\mathcal{F}_t]$, where $\mathcal{F}_t$ is the filtration generated by $y_1, \ldots, y_t$.

Smoothing

On the one hand,

$$\begin{aligned}&\mathbb{E}[\mathbb{V}[X_{n-1}|X_n, y_{1:n-1}]|y_{1:n}]\\&= \mathbb{E}[C_{n-1|n-1} - L_{n-1}C_{n|n-1}L_{n-1}^T|y_{1:n}]\\&= C_{n-1|n-1} - L_{n-1}C_{n|n-1}L_{n-1}^T.\end{aligned}$$

On the other hand,

$$\begin{aligned}&\mathbb{V}[\mathbb{E}[X_{n-1}|X_n, y_{1:n-1}]|y_{1:n}]\\&= \mathbb{V}[m_{n-1|n-1} + L_{n-1}(X_n - m_{n|n-1})|y_{1:n}]\\&= L_{n-1}\mathbb{V}[X_n|y_{1:n}]L_{n-1}^T\\&= L_{n-1}C_{n|n}L_{n-1}^T.\end{aligned}$$

So

$$\mathbb{V}[X_{n-1}|y_{1:n}] = C_{n-1|n-1} + L_{n-1}\left(C_{n|n} - C_{n|n-1}\right)L_{n-1}^T.$$

Smoothing

To summarize,

$$
\begin{aligned}
m_{n-1|n} &= m_{n-1|n-1} + L_{n-1}(m_{n|n} - m_{n|n-1}),\\
C_{n-1|n} &= C_{n-1|n-1} + L_{n-1}\left(C_{n|n} - C_{n|n-1}\right) L_{n-1}^T.
\end{aligned}
$$

and more generally, with $L_{t-1} = C_{t-1|t-1}\Phi^T C_{t|t-1}^{-1}$,

$$
\begin{aligned}
m_{t-1|n} &= m_{t-1|t-1} + L_{t-1}(m_{t|n} - m_{t|t-1}),\\
C_{t-1|n} &= C_{t-1|t-1} + L_{t-1}\left(C_{t|n} - C_{t|t-1}\right) L_{t-1}^T.
\end{aligned}
$$

Compute all smoothing quantities in one backward pass, given the outputs of the forward pass.

Computational cost in $\mathcal{O}(n)$.

Smoothing: summary

- Given $y_1, \dots, y_{t-1}$, compute distribution of $X_{t-1} | X_t = x_t$.
- We do not know x_t, but we know $X_t \sim \mathcal{N}(m_{t|n}, C_{t|n})$, given all the observations.
- Retrieve the distribution of X_{t-1} given all the observations, using tower property and Eve's law:

$$\begin{aligned}
\mathbb{E}[X_{t-1}|y_{1:n}] &= \mathbb{E}[\mathbb{E}[X_{t-1}|X_t, y_{1:t-1}]|y_{1:n}], \\
\mathbb{V}[X_{t-1}|y_{1:n}] &= \mathbb{E}[\mathbb{V}[X_{t-1}|X_t, y_{1:t-1}]|y_{1:n}] \\
&\quad + \mathbb{V}[\mathbb{E}[X_{t-1}|X_t, y_{1:t-1}]|y_{1:n}].
\end{aligned}$$

Filtering and smoothing: implementation

- Can be implemented manually.

  Matrix multiplication `%*%`, system solving/inverse `solve`.

- In R, functions `Kfilter0`, `Ksmooth0` from package `astsa`.

  Functions `dlmModPoly`, `dlm`, `dlmFilter`, `dlmSmooth`, from package `dlm`.

- Readily available implementations in most programming languages.

Missing observations

We might only observe Y at non-equispaced times $t_1, \ldots, t_n$.

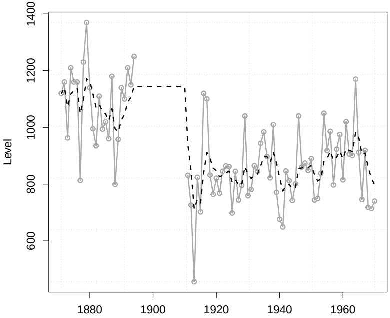
We can still define the latent process (X_t) at all times.

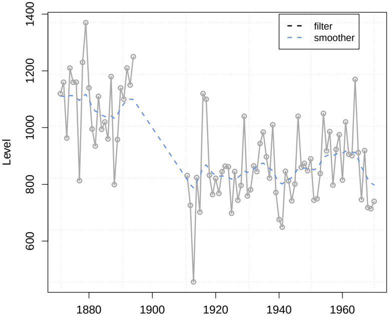
We can adapt the Kalman filter easily, by skipping the update steps at the times for which no observation is available.

The ability to handle missing observations easily is an appeal of state space modeling.

Missing observations

Nile river annual flow at Aswan, 1871–1970.





Sampling paths

Limitation of the Kalman smoother

The Kalman smoother provides $m_{t|n}$ and $C_{t|n}$, which are "marginal quantities". Can be used to compute e.g.

$$\mathbb{P}(X_t > c | Y_1, \ldots, Y_n).$$

As opposed to "joint quantities", which would involve the distribution of the entire trajectories $(X_0, \ldots, X_n)$ given $(Y_1, \ldots, Y_n)$.

For instance, we might be interested in quantities such as

$$\mathbb{P}(\{\exists t \in \{0, \ldots, n\}\ X_t > c\} | Y_1, \ldots, Y_n).$$

Sampling paths

Thankfully, we can sample paths $X_{0:n}$ given $Y_{1:n}$.

With $L_{t-1} = C_{t-1|t-1}\Phi^T C_{t|t-1}^{-1}$,
$X_{t-1}|X_t = x_t, y_1, \ldots, y_n$ follows a normal distribution with

$$\begin{aligned} \text{mean} &= m_{t-1|t-1} + L_{t-1}(x_t - m_{t|t-1}), \\ \text{variance} &= C_{t-1|t-1} - L_{t-1} C_{t|t-1} L_{t-1}^T. \end{aligned}$$

Thus we can sample $X_n \sim \mathcal{N}(m_{n|n}, C_{n|n})$, and recursively $X_{t-1}|X_t = x_t, y_1, \ldots, y_n$ following the above formula.

Beyond linear Gaussian models

The tasks of filtering, likelihood calculations, and smoothing, can be considered for nonlinear models (they are well-defined).

However, the Kalman recursions rely heavily on properties of multivariate Normal distributions, which are lost when

- the equations are non-linear or
- when the noise terms are non-Gaussian.

Examples: stochastic volatility model, *Markov-switching models*

We can approximate these recursions with Monte Carlo samples instead of exact distributions.

This leads to “particle filters” and “particle smoothers”.

Take aways

- State space models in their most generic form encompass a wide range of time series models, have an intuitive interpretation, can handle missing observations.

- Linear Gaussian models include ARMA and structural time series models, and computation can be performed with the Kalman filter.

- Nonlinear or non-Gaussian models, Markov-switching models etc, are used in various settings, but the computations are significantly more challenging.

- Trade-offs between modeling flexibility and computational feasibility are commonplace in data analysis; state space models provide an illustration.