Time Series Models 2021
Assignments

Siem Jan Koopman Janneke van Brummelen Karim Moussa Ilka van de Werve

Last update: March 1, 2021

Introduction

- Both assignments are collected in this document
- This document may be updated throughout the course, so always make sure to have the latest version
- You work in groups of 4 students (make sure to enroll before Friday February 5 at 23h59 via `https://forms.gle/YtweWyubpmfBAHhf7`; incomplete groups should register before Sunday February 7 at 23h59 via `https://forms.gle/48qsLZ6JEvZwdre87`)
- Students who did the assignments last year, are advised to contact Ilka van de Werve (`i.vande.werve@vu.nl`)
- **Deadlines:** Friday February 19 at 23h59 (Assignment 1, extension incorporated) and Wednesday March 10 at 23h59 (Assignment 2)
- As a group, hand in the solutions (.pdf and code) via Canvas Assignments
- You can use "any" programming language but no packages: best support can be provided for the programming platforms of `Python`, `R`, `Matlab`, and `Ox`.
- Data from the DK-book can be found at `http://www.ssfpack.com/DKbook.html`
- Support for the assignments is given by Karim Moussa: use Canvas Discussions for assignment questions (you can typically expect an answer within two working days and don't share assignment code here, just briefly explain what is going on) or use the weekly office hours for coding questions
- Based on your input via Canvas Discussions, Ilka van de Werve will give tips in the tutorials as well

Good luck!

Assignment 1

- Assignment 1 is about implementing the methods for the Local Level model and about interpreting your results
- The computer code for this assignment will also form the basis for Assignment 2, so write "clean" code
- First, you will replicate almost all figures in Chapter 2 of the DK-book (for the Nile data) to ensure and verify that your code works correctly
- Second, you will analyze a data set of your choice using the same model

Consider Chapter 2 of the DK-book, there are 8 figures for the Nile data

(a) Write computer code that can reproduce all these figures except Figure 2.4

(b) Implement it according to the set of recursions for the local level model

- Please note that these equations are a result of declaring $Z_t = T_t = R_t = 1$, $d_t = c_t = 0$, $H_t = \sigma_\varepsilon^2$ and $Q_t = \sigma_\eta^2$ in the general state space model (no need to implement methods for the general case here, but it shows you the link);
- Figures 2.3 (ii),(iv) plot standard deviations instead of variances

(c) Write a short documentation of your computer code and explain shortly what you obtained in the different figures and how they relate to each other

(d) Use a time series data-set of your choice (provide the source), motivate why the local level model is appropriate for your choice of time series, and show your output (the same figures as for the Nile data), and carefully discuss your results

Assignment 2

- Assignment 2 is for the Stochastic Volatility (SV) model
- We will make some simplifying assumptions and make a start with the analysis of the Stochastic Volatility model using linear Kalman filter methods
- You can proceed with your code of the first assignment and use return data from the DK-book

Background information

Denote closing price at trading day t by P_t with its return

$$r_t = \log(P_t \,/\, P_{t-1}) = \Delta \log P_t = \Delta p_t, \qquad t = 1, \ldots, n.$$

The price p_t can be regarded as a discretized realisation from a continuous-time log-price process $\log P(t)$, that is

$$\mathrm{d} \log P(t) = \mu \, \mathrm{d}\, t + \sigma(t) \, \mathrm{d}\, W(t),$$

where μ is the mean-return, $\sigma(t)$ is a continuous volatility process and $W(t)$ is standardised Brownian motion. We concentrate on the volatility process and we let $\log \sigma(t)^2$ follow a so-called Ornstein-Uhlenbeck process

$$\log \sigma(t)^2 = \xi + H(t), \qquad \mathrm{d}\, H(t) = -\lambda H(t) \, \mathrm{d}\, t + \sigma_\eta \, \mathrm{d}\, B(t),$$

where ξ is constant, $0 < \lambda < 1$, σ_η is the "volatility-of-volatility" coefficient (strictly positive) and $B(t)$ is standardised Brownian motion, independent of $W(t)$.

The general framework can lead to a statistical model for the daily returns y_t. By applying the Euler-Maruyama discretisation method, we obtain the SV-model as

$$y_t = \mu + \sigma_t \varepsilon_t, \qquad \log \sigma_t^2 = \xi + H_t, \qquad H_{t+1} = \phi H_t + \sigma_\eta \eta_t, \tag{1}$$

where $\phi = 1 - \lambda$ so that $0 < \phi < 1$. Since both σ_t and ε_t are stochastic processes, we have a nonlinear time series model.

However, after data transformation $x_t = \log(y_t - \mu)^2$ and some redefinitions, we obtain

$$x_t = h_t + u_t, \qquad h_{t+1} = \omega + \phi h_t + \sigma_\eta \eta_t, \tag{2}$$

where $u_t = \log \varepsilon_t^2$, $\omega = (1-\phi)\xi$ and $h_t = H_t + \xi$.

We obtain the linear AR(1)+noise model, but the disturbance u_t is not necessarily Gaussian. This is the basis of quasi maximum likelihood (QML) for the SV-model. When we assume ε_t is Gaussian, u_t is generated from a $\log \chi^2$ distribution from which the mean and variance are well-defined (see hint in question (c)).

The QML-method adopts the Kalman filter to compute the likelihood; do as if u_t is Gaussian with mean and variance corresponding to those of the $\log \chi^2$ distribution. This analysis can be regarded as an approximate analysis.

- Use the SV-data of the DK-book

  (a) Make sure that the financial series is in returns (transform if needed, see Figure 14.5). Present graphs and descriptives.

  (b) The SV-model can be made linear by transforming the returns data to $x_t = \log y_t^2$. This is the basis of the QML-method. Compute x_t and present a graph. Hint: avoid taking logs of zeros, you can do so by demeaning y_t if needed.

  (c) The disturbances in the model for x_t will not be Gaussian distributed. But we can assume that they are Gaussian with mean and variance corresponding to those of the $\log \chi^2(1)$ distribution. Using equation (2), estimate the unknown coefficients by the QML-method using the Kalman filter and present the results in a Table. Hint: the $\log \chi^2(1)$ distribution has mean -1.27 and variance $\pi^2/2 = 4.93$ (mean adjustment and fixed variance).

  (d) Take the QML-estimates as your final estimates. Compute the smoothed values of h_t based on the approximate model for x_t in equation (2) by using the Kalman filter and smoother, and present them in a graph along with the transformed data x_t. In addition, present both the *filtered* ($\mathbb{E}[H_t|Y_t]$) and *smoothed* ($\mathbb{E}[H_t|Y_n]$) estimates of H_t in a graph.

(e) Extension 1: For a period of at least five years, consider the daily returns for S&P500 index (or another stock index) that you can obtain from Yahoo Finance `https://finance.yahoo.com/lookup`. You can re-visit the analysis of above. To improve the performance of the SV model, you can extend your analysis with a Realized Volatility measure which can be obtained from Oxford-Man Institute `https://realized.oxford-man.ox.ac.uk/`.

For this purpose, you can consider the extended model

$$x_t = \beta \cdot \log \mathrm{RV}_t + h_t + u_t, \qquad h_{t+1} = \omega + \phi h_t + \sigma_\eta \eta_t, \tag{3}$$

where β is the regression coefficient and RV_t is the realized volatility measure of your choice. How does the analysis above change with the inclusion of RV ? Implement the procedure and interpret your results.

(f) Extension 2: We return to the original SV model of above (so without RV), that is equation (1), the nonlinear expression of the SV model. Compute the filtered estimates of H_t in equation (1) using the particle filter and compare it with the earlier *filtered* QML estimates of H_t in a graph. You can do this for the original data set and repeat it for the stock index of Extension 1.