Learning Models with Simulation: Project 2

Instructions

Answer all questions. Save the Python code you have written as a Python script or Jupyter Notebook. All plots and discussion could be included in the Jupyter Notebook or be saved separately as a document. Submit your solutions on our Moodle course page.

State space modeling for the annual flow of river Nile

The first part of this project considers the use of state space models for hydrology applications. The dataset `nile.csv` contains measurements of the annual flow (in unit of $10^8 \times m^3$) of the river Nile at Aswan between 1871 to 1970 (source: Table 1 of Cobb 1978).

1. Read the river Nile dataset using the `read_csv` function from the pandas package.

2. We will use the first 80 measurements from 1871 to 1950 as our training dataset and the last 20 measurements from 1951 to 1970 as our testing dataset. Implement this split.

3. Plot the time series and observe the apparent changepoint near 1898. Speculate why this might be the case.

4. We will model the annual flow X_t and its measurement Y_t in year $t+1871$, for $t = 0, 1, \dots, 99$, using the following univariate linear Gaussian state space model

$$\begin{aligned} &X_0 \sim N(1120, 1450), \\ &X_t = X_{t-1} + U_t, \quad U_t \sim N(0, \sigma_X^2), \\ &Y_t = X_t + V_t, \quad V_t \sim N(0, \sigma_Y^2). \end{aligned} \tag{1}$$

   Using the `kalman` module from the `particles` package, write a function that constructs the state space model (1) for any choice of parameters $\theta = (\sigma_X^2, \sigma_Y^2)$. This function should take σ_X^2 and σ_Y^2 as arguments and output a `MVLinearGauss` object.

5. Let $y_0, y_1, \dots, y_{79}$ denote the 80 measurements from 1871 to 1950 in our training dataset. Using the `kalman` module from the `particles` package, write a function that evaluates the log-likelihood $\log p(y_0, y_1, \dots, y_{79}|\theta)$ for any $\theta \in (0, \infty)^2$ with the Kalman filter. This function should take as argument `theta` a vector of size 2 and output a numerical value.

6. Compute the maximum likelihood estimator

$$\widehat{\theta} = \arg\max \log p(y_0, y_1, \ldots, y_{79}|\theta)$$

using the `minimize` function from the SciPy `optimize` subpackage. Initialize the optimization routine at $\theta = (\sigma_X^2, \sigma_Y^2) = (1450, 15000)$ and use the Nelder–Mead algorithm by specifying the `method` parameter.

7. Using the `kalman` module from the `particles` package, perform Kalman filtering and Kalman smoothing on the state space model (1) with the maximum likelihood estimator $\widehat{\theta}$.

8. Plot the filtering mean $E[X_t|\widehat{\theta}, y_0, y_1, \ldots, y_t]$, the smoothing mean $E[X_t|\widehat{\theta}, y_0, y_1, \ldots, y_{79}]$ and the measurement y_t for $t = 0, 1, \ldots, 79$. Comment on the differences between the filtering and smoothing means. Explain why the changepoint near 1898 is also reflected in these means.

9. Compare the filtering variance $\mathrm{Var}[X_t|\widehat{\theta}, y_0, y_1, \ldots, y_t]$ and the smoothing variance $\mathrm{Var}[X_t|\widehat{\theta}, y_0, y_1, \ldots, y_{79}]$ for $t = 0, 1, \ldots, 79$. Explain these differences.

10. Using the state space model (1) with the maximum likelihood estimator $\widehat{\theta}$, compute the predictive mean $E[Y_t|\widehat{\theta}, y_0, y_1, \ldots, y_{79}]$ and the predictive variance $\mathrm{Var}[Y_t|\widehat{\theta}, y_0, y_1, \ldots, y_{79}]$, for $t = 80, \ldots, 99$, of the 20 measurements from 1951 to 1970.

11. Using the predictive means and variances, compare the predictive distributions with the 20 measurements from 1951 to 1970 in our testing dataset.

Bass diffusion model for the number of YouTube users

The second part of this project considers the use of Bass diffusion models to analyze the rise of social media. The dataset `youtube.csv` contains estimates of the number of monthly active YouTube users since the company started in 2005 to 2018 (source: Statista and The Next Web). You should work on the Jupyter Notebook `youtube.ipynb`, which imports the necessary packages and modules in the first cell, and creates a specific object for the shifted binomial distribution in the second cell. We will write $\mathrm{ShiftedBinomial}(n, p, s)$ to denote the distribution of $Y = s + X$ if $X \sim \mathrm{Binomial}(n, p)$ for any $s \in \{0, 1, \ldots\}$. The notation $\mathrm{TruncNormal}(\mu, \sigma^2)$ will refer to the Normal distribution $N(\mu, \sigma^2)$ truncated to the set of positive real numbers $(0, \infty)$.

1. Read the YouTube dataset using the `read_csv` function from the pandas package.

2. Plot the time series and comment on its behaviour.

3. We will model the number of users on YouTube X_t and its measurement Y_t in year $t + 2005$, for $t = 0, 1, \ldots, 13$, using the following stochastic Bass state space model

$$\begin{aligned} X_0 &\sim \mathrm{Binomial}(N, \beta_0), \\ X_t &\sim \mathrm{ShiftedBinomial}(N - X_{t-1}, \alpha + \beta X_{t-1}/N, X_{t-1}), \\ Y_t &\sim \mathrm{TruncNormal}(X_t, \sigma^2), \end{aligned} \tag{2}$$

with unknown parameters $\theta = (\beta_0, \alpha, \beta, \sigma)$. We will set N as the world population size of 7.7 billion. Using the `particles` package, define the Bass state space model (2) as a class `bass` with methods `PX0`, `PX` and `PY`. [Hint: use the `ShiftedBinomial` object and refer to the documentation page `https://particles-sequential-monte-carlo-in-python.readthedocs.io/en/latest/distributions.html`.]

4. We adopt the following independent prior distribution $p(\theta)$ for the parameters

$$\beta_0, \alpha \sim \text{Beta}(1, 100), \quad \beta \sim \text{Beta}(2, 20), \quad \sigma \sim \text{TruncNormal}(1.5 \times 10^8, (0.3 \times 10^8)^2). \tag{3}$$

   Using the `particles` package, define the prior distribution in (3). Plot the prior probability density function of each parameter and comment on the appropriateness of this prior distribution. [Hint: use the method `logpdf` that distribution objects have and use the interpretation of the parameters in (2).]

5. Run a particle marginal Metropolis–Hastings with 20 particles for 5000 iterations to sample from the posterior distribution $p(\theta|y_0, y_1, \ldots, y_{13})$. [Warning: this may take a minute on your machine.] Report the resulting acceptance rate. Use diagnostic plots to examine the mixing properties of the Markov chain and an appropriate choice of burn-in.

6. Using your choice of burn-in, approximate the posterior mean of the parameters $E[\theta|y_0, y_1, \ldots, y_{13}]$. Let $\widehat{\theta}$ denote the resulting approximation.

7. Using the `particles` package, run a bootstrap particle filter with 1000 particles on the Bass state space model (2) with the approximate posterior mean $\widehat{\theta}$. Examine the effective sample sizes and comment on the performance of the particle filter.

8. Plot the observation y_t and the particle filter approximation of the filtering mean $E[X_t|\widehat{\theta}, y_0, y_1, \ldots, y_t]$ for $t = 0, 1, \ldots, 13$. Comment on your findings.

9. Using the `particles` package, run forward filtering and backward sampling to obtain 1000 samples from the smoothing distribution $p(x_0, x_1, \ldots, x_{13}|\widehat{\theta}, y_0, y_1, \ldots, y_{13})$. Approximate the smoothing means $E[X_t|\widehat{\theta}, y_0, y_1, \ldots, y_{13}]$, for $t = 0, 1, \ldots, 13$, and compare them to your approximation of the filtering means. Comment on your findings.