Sinusoidal model

Adeline Leclercq Samson

2023-01-11

Description of the sinusoidal model

The observations along the tusk are denoted Y_i for $i = 1, \ldots, n$, with the corresponding position along the tusk denoted x_i. The model is the following

$$Y_i = f(x_i, \varphi) + \varepsilon_i$$

with ε_i a random noise assumed to be normally distributed with mean 0 and variance ω^2. The regression function is a periodic sinusoidal function

$$f(x, \varphi) = A \sin(g(x) + b) + B \sin(2g(x) + 2b + \pi/2)$$

with function g defined as

$$g(x) = ax + \xi_x$$

and finally ξ_x is assumed to be a random Ornstein-Uhlenbeck process

$$d\xi_x = -\beta \xi_x dx + \sigma dW_x$$

The objective is to estimate the parameters $\varphi = (A, B, a, b)$; $\psi = e^{\beta\Delta}$ where Δ is the step size between two observations and $\gamma^2 = \frac{\sigma^2}{2\beta}(1 - e^{-2\beta\Delta})$.

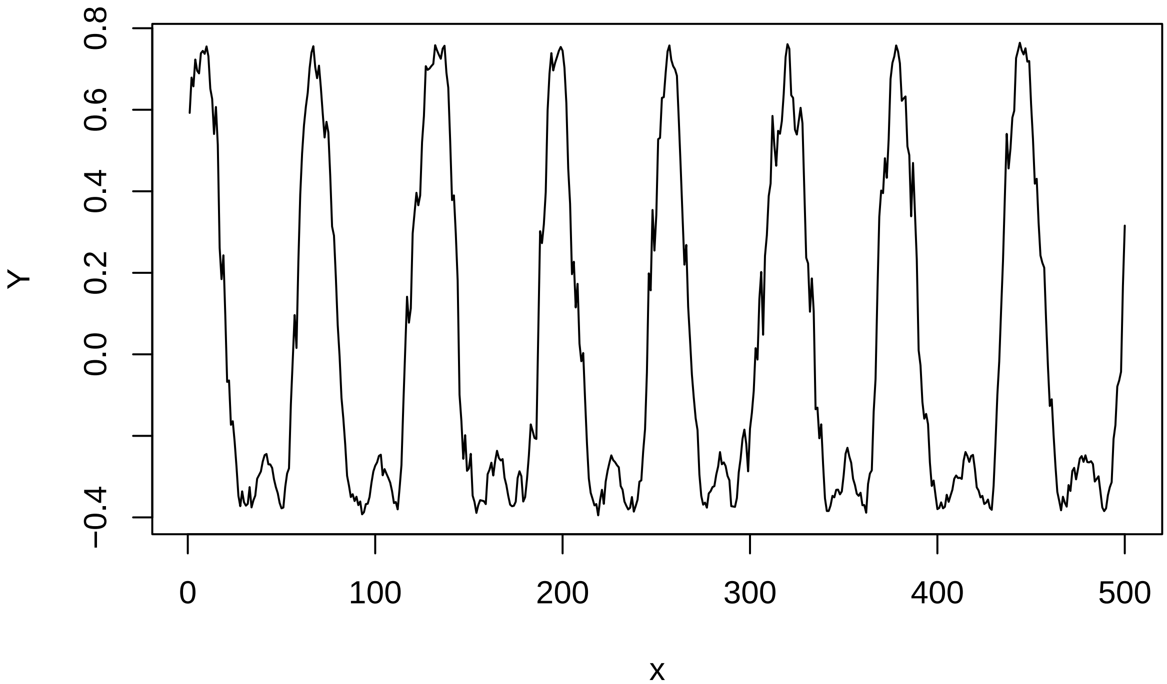
Simulation of trajectories

Let us start with some simulations.

```
# parameters
A = 1/2
B = -1/4
b = 1
a = 0.1
beta = 0.05
sigma = 0.1
omega = 0.01

# simulations
n = 500
x = 1:n
delta = 1
psi = exp(-delta*beta)
gamma = sigma/sqrt(2*beta)*sqrt(1-psi^2)
xi = rep(0,n)
for (i in 2:n){
  xi[i] = xi[i-1]*psi + rnorm(1,0, gamma)
}
```

```
gx = a*x+xi
fx = A*sin(gx+b) + B*sin(2*gx + 2*b + pi/2)
Y = fx + rnorm(n,0,omega)
plot(x,Y,type = "l")
```



Estimation with the EM algorithm

The EM algorithm is based on the complete log-likelihood of the model. The solution of the hidden process ξ_x is

$$\xi_{x+\Delta} = \xi_x e^{-\beta\Delta} + \int_x^{x+\Delta} \sigma e^{\beta(s-x)} dW_s$$

such that the transition density is

$$p(\xi_{x+\delta}|\xi_x) = \mathcal{N}(\xi_x e^{-\beta\Delta}, \frac{\sigma^2}{2\beta}(1-e^{-2\beta\Delta}))$$

The complete log-likelihood is thus

$$\begin{aligned}
\log L(Y,\xi,\theta) &= \sum_{i=1}^{n} \log p(Y_i|\xi_i) + \sum_{i=1}^{n} \log p(\xi_i|\xi_{i-1}) + \log p(\xi_1) \\
&= -\sum_{i=1}^{n} \frac{(Y_i - f(x_i,\varphi))^2}{2\omega^2} - \frac{n}{2}\log(\omega^2) \\
&\quad -\sum_{i=1}^{n} \frac{(\xi_i - \xi_{i-1}e^{-\beta\Delta})^2}{\frac{\sigma^2}{\beta}(1-e^{-2\beta\Delta})} - \frac{n}{2}\log(\frac{\sigma^2}{2\beta}(1-e^{-2\beta\Delta})) \\
&= -\sum_{i=1}^{n} \frac{(Y_i - f(x_i,\varphi))^2}{2\omega^2} - \frac{n}{2}\log(\omega^2) \\
&\quad -\sum_{i=1}^{n} \frac{(\xi_i - \xi_{i-1}\psi)^2}{2\gamma^2} - \frac{n}{2}\log(\gamma^2)
\end{aligned}$$

The EM algorithm proceeds at iteration k with the two following steps, given the current value of the parameters θ^k

- E step: calculation of $Q(\theta, \theta^k)$
- M step: update of the parameters $\theta^{k+1} = \arg\max_\theta Q(\theta, \theta^k)$

E step The condition distribution $p(\xi|Y;\theta^k)$ is not explicit because the regression function is not linear. We should proceed with a MCMC algorithm to sample from this distribution. This will lead to a stochastic version of the EM algorithm, namely the SAEM algorithm.

M step We need the sufficient statistics to update the algorithm.

The statistics are

$$\begin{aligned}
S_1(\xi_i) &= \frac{1}{n}\sum_{i=1}^{n}(Y_i - f(x_i(\xi_i), \varphi))^2 \\
S_2(\xi_i) &= \sum_{i=1}^{n}\xi_{i-1}\xi_i \\
S_3(\xi_i) &= \sum_{i=1}^{n}\xi_{i-1}^2 \\
S_4(\xi_i) &= \sum_{i=1}^{n}\xi_i^2
\end{aligned}$$

The update of the parameters are based on these statistics.

SAEM algorithm The steps of the SAEM algorithm are

- E step: simulation of a new trajectory ξ^k with a MCMC algorithm targeting $p(\xi|Y;\theta^k)$ as stationary distribution
- SA step: stochastic approximation of the sufficient statistics

$$\begin{aligned}
s_1^k &= s_1^{k-1} + (1-\alpha_k)(S_1(\xi^k) - s_1^{k-1}) \\
s_2^k &= s_2^{k-1} + (1-\alpha_k)(S_2(\xi^k) - s_2^{k-1}) \\
s_3^k &= s_3^{k-1} + (1-\alpha_k)(S_3(\xi^k) - s_3^{k-1}) \\
s_4^k &= s_4^{k-1} + (1-\alpha_k)(S_4(\xi^k) - s_4^{k-1})
\end{aligned}$$

- M step: update of θ^k using the sufficient statistics s^k

$$\begin{aligned}
\widehat{\varphi}^k &= \arg\min_\varphi \sum_{i=1}^{n}\left(y_i - f(x_i(\xi_i^k), \varphi)\right)^2 \\
\widehat{\psi}^k &= \frac{s_2^k}{s_3^k} \\
\widehat{\omega^2}^k &= s_1^k \\
\widehat{\gamma^2}^k &= \frac{1}{n}(\widehat{\psi^2}^k s_3^k - 2\widehat{\psi}^k s_2^k + s_4^k)
\end{aligned}$$

MCMC On fixe le jeu de données

```
xi_true = xi
```

On initialise l'algo

```
xi_init = rep(0,n)
for (i in 2:n){
  xi_init[i] = xi_init[i-1]*psi + rnorm(1,0, gamma)
}
```