Bayesian Scientific Computing
Exercise for Day 5

Daniela Calvetti, Erkki Somersalo

Case Western Reserve University
Department of Mathematics, Applied Mathematics and Statistics

Lyngby, December 2019

Problem description

- Consider an object such as a meteorite or a space module entering the atmosphere.
- The object is monitored by a radar that measures at discrete times the object's distance from the ground.
- The problem we consider here is to estimate the object's velocity and the ballistic coefficient.

Model

$$
\begin{aligned}
h &= h(t) = \text{altitude of the object}, \quad h > 0, \\
v &= v(t) = \text{downwards speed}, \quad v > 0.
\end{aligned}
$$

Equation of motion:

$$\frac{dh}{dt} = -v. \tag{1}$$

The acceleration depends on two factors:

1. the Earth's gravitational field,
2. the drag of the atmosphere.

Model

Differential equation:

$$\frac{dv}{dt} = g - g\frac{\rho(h)\, v^2}{2\beta}, \tag{2}$$

where

$$\begin{aligned} g &= \text{constant acceleration due to the gravity} = 9.81\,\mathrm{m/s^2}, \\ \rho(h) &= \text{altitude-dependent density of the air}, \\ \beta &= \text{ballistic constant}, \quad \beta > 0. \end{aligned}$$

The unknown ballistic constant depends on the shape, mass, and cross-sectional area of the object.
To express the belief that β is constant, we write

$$\frac{d\beta}{dt} = 0. \tag{3}$$

Model

For the atmospheric density, we assume that it has a known dependency on h, given by

$$\rho(h) = \gamma e^{-\eta h}, \tag{4}$$

where $\gamma = 1.754\,\mathrm{kg/m^3}$ and $\eta = 1.39 \times 10^{-4}\,\mathrm{m^{-1}}$.
We define the vector of unknowns as

$$x(t) = \begin{bmatrix} h(t) \\ v(t) \\ \beta \end{bmatrix}.$$

The equations (1)–(4) define a non-linear system

$$\frac{dx}{dt} = f(x) = \begin{bmatrix} -v \\ g - g\,\frac{\rho(h)\,v^2}{2\beta} \\ 0 \end{bmatrix}. \tag{5}$$

Model

Observation model: A ssume that at times $t_j = j\Delta t$, $j = 0, 1, 2, \ldots$, the variable h is observed.
The observation model is

$$b_j = q^{\mathsf{T}} x(t_j) + \varepsilon_j,$$

where

$$q = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2).$$

Discretized propagation model: Forward Euler,

$$x(t_{j+1}) \approx x(t_j) + \Delta t\, f(x(t_j)).$$

Bayesian framework: Model the discretization error, error in the model etc. as a random innovation:

$$x(t_{j+1}) = \underbrace{x(t_j) + \Delta t f(x(t_j))}_{=F(x(t_j))} + v_j = F(x(t_j)) + v_{j+1}, \quad v_{j+1} \sim \mathcal{N}(0, \mathsf{C}).$$

The covariance matrix $\mathsf{C} \in \mathbb{R}^{3\times 3}$ is assumed to be diagonal.

Problem formulation

Problem: *Given the observations b_j, $j = 0, 1, \ldots$ estimate the state vectors $x(t_j)$, $j = 0, 1, \ldots$.*

Generate data: Assume that the target has the initial altitude h_0, initial velocity v_0, and the true ballistic coefficient β_*,

$$
\begin{aligned}
h_0 &= 61\,000\,\mathrm{m} \\
v_0 &= 3\,048\,\mathrm{m/s}, \\
\beta_* &= 19\,161\,\mathrm{kg/ms^2}.
\end{aligned}
$$

To generate the noiseless data, use the ODE solver `ode45`. You need to write a right hand side function

```
function rhs = TrackingDynamics(t,x)
```

Problem formulation

Compute the noiseless data at $t_j = j\Delta t$, $j = 0, 1, \ldots, 300$ with $\Delta t = 0.1\,\mathrm{s}$, that is, we assume that the object is monitored for half a minute.
After generating the noiseless vectors $x(t_j) \in \mathbb{R}^3$, extract the first component $x_1(t_j)$ (altitude) and add to it Gaussian noise

$$b_j = x_1(t_j) + \varepsilon_j, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2),$$

where you can use $\sigma = 500\,\mathrm{m}$. Now you have the data

$$b_0, b_1, \ldots, b_T, \quad T = 300.$$

Problem formulation

Generate the initial particle cloud: As initial guess for the altitude, use the first radar reading

$$h_{\text{init}} = b_0.$$

As initial guess for the velocity and ballistic coefficient, use

$$\begin{aligned} v_{\text{init}} &= 3\,000\,\mathrm{m/s}, \\ \beta_{\text{init}} &= 20\,000\,\mathrm{kg/ms}^2. \end{aligned}$$

Set the number of particles to $N = 5\,000$ be the number of the particles.

Problem formulation

Generate the initial sample by

$$x^{(\ell)}(t_0) = x^{(\ell)} = \begin{bmatrix} h_{\mathrm{init}} \\ v_{\mathrm{init}} \\ \beta_{\mathrm{init}} \end{bmatrix} + \mathsf{C}^{1/2} w^{(\ell)}, \quad w^{(j)} \sim \mathcal{N}(0, \mathsf{I}_3),$$

where $1 \leq \ell \leq N$, the prior covariance matrix C is diagonal. For the prior standard deviations (diagonal entries of $\mathsf{C}^{1/2}$), use the values

$$\begin{aligned} STD_h &= 500\,\mathrm{m}, \\ STD_v &= 200\,\mathrm{m/s}, \\ STD_\beta &= 1\,500\,\mathrm{kg/ms^2}. \end{aligned}$$

Assign equal weight $w^{(\ell)} = 1/N$ to each particle.

Problem formulation

Build the particle filter algorithm. Here is an outline: For each time t_j, $j = 0, 1, \ldots, T$,

(a) Propagate each particle $x^{(\ell)} = x^{(\ell)}(t_j)$ using the Forward Euler, to have the new candidate particles $\widehat{x}^{(\ell)}$ for the next time instance.

(b) For every candidate particle, compute the fitness weight,

$$\widehat{w}^{(\ell)} = w^{(\ell)} \times \exp\left(-\frac{1}{2\sigma^2}|\widehat{x}_1^{(\ell)} - b_{j+1}|^2\right), \quad \widehat{w}^{(\ell)} = \frac{\widehat{w}^{(\ell)}}{\sum \widehat{w}^{(\ell)}}.$$

Problem formulation

(c) Using the fitness weights as probabilities, draw with replacement N indices $k_\ell \in \{1, 2, \ldots, n\}$, and generate the new particles by adding innovation,

$$x_{\rm new}^{(\ell)} = \widehat{x}^{(k_\ell)} + \mathsf{D}^{1/2} w^{(\ell)}, \quad w^{(\ell)} \sim \mathcal{N}(0, \mathsf{I}_3)$$

where $\mathsf{D} \in \mathbb{R}^{3\times 3}$ is the covariance matrix of the innovation. You may assume that the innovation covariance matrix is diagonal, and the standard deviations of the innovation are

$$\begin{aligned} STD_h &= 100\,\mathrm{m}, \\ STD_v &= 100\,\mathrm{m/s}, \\ STD_\beta &= 5\,\mathrm{kg/ms^2}. \end{aligned}$$

Problem formulation

(d) Update the weights, setting

$$w^{(\ell)} = \exp\left(-\frac{1}{2\sigma^2}\left(|(x_{\rm new}^{(\ell)})_1 - b_{j+1}|^2 - |x_1^{(k_\ell)} - b_{j+1}|^2\right)\right), \quad w^{(\ell)} = \frac{w^{(\ell)}}{\sum w^{(\ell)}}.$$

(e) Update $x^{(\ell)} = x_{\rm new}^{(\ell)}$, $j \to j+1$, and continue.

Problem formulation

Once your particle algorithm is operational, the time of show and tell begins! At each time instant t_j calculate and visualize the mean value of the particle cloud, the median, and credibility intervals of p % for each of the three variables. Here is how you do the latter: Suppose that at time t_j, the altitudes corresponding the particles are given by $h_1, h_2, \ldots, h_N$. After sorting the values in an increasing order we have

$$h_1 \leq h_2 \leq \cdots \leq h_N.$$

The interval that contains p% of the particles is found by throwing away symmetrically the same amount of the smallest and the largest values. The credibility interval is given by $[h_{i_{\min}}, h_{i_{\max}}]$, where

$$i_{\min} = [(1 - p/100)(N/2)], \quad i_{\max} = [(1 + p/100)(N/2)],$$

where the brackets mean "rounding to the nearest integer".