# Sequential Importance Sampling (SIS)

Sensor fusion & nonlinear filtering

Lars Hammarstrand

- Objective: to recursively and accurately approximate the filtering density, $p(\mathbf{x}_k | \mathbf{y}_{1:k})$.
- Assumption: both the motion and measurement models

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}) \qquad \text{and} \qquad p(\mathbf{y}_k | \mathbf{x}_k)$$

can be easily evaluated point-wise.

- A common example is

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1}, \quad & \mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}) \\ \mathbf{y}_k &= h(\mathbf{x}_k) + \mathbf{r}_k & \mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \end{aligned}$$

where, e.g., $p(\mathbf{x}_k | \mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k; f(\mathbf{x}_{k-1}), \mathbf{Q}_{k-1})$ is generally easy to evaluate for any values of $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$.

- Particle filters are also known as *sequential importance resampling* or *sequential Monte Carlo*.
- The basis of these methods is an algorithm called sequential importance sampling (SIS).

**Standard SIS algorithm**

- For $i = 1, \ldots, N$ and at each time $k$:
  - Draw $\mathbf{x}_k^{(i)} \sim q(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)$.
  - Compute weights

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)}.$$

  - Normalize the weights.

- We then approximate $p(\mathbf{x}_k | \mathbf{y}_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)})$.

- Assuming that we describe our posterior using the following approximation $p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)})$. What is then the MMSE estimate of $\mathbf{x}_k$?

  - $\hat{\mathbf{x}}_k = \sum_i^N w_k^{(i)} \mathbf{x}_k^{(i)}$
  - $\hat{\mathbf{x}}_k = \mathbf{x}_k^{(j)}$, where $j = \arg\max_i w_k^{(i)}$
  - $\hat{\mathbf{x}}_k = \frac{1}{N} \sum_i^N \mathbf{x}_k^{(i)}$
  - It is not possible to calculate a MMSE estimate from this approximation.

$$\hat{x}_k = E\{x_k | y_k\} = \int x_k \sum_{i=1}^{N} \omega_k^{(i)} \delta(x_k - x_k^{(i)}) dx$$

$$= \sum_{i=1}^{N} \omega_k^{(i)} x_k^{(i)}$$

**Derivation - Basic strategy**

Recursively at time $k = 1, 2, \dots$

1. Draw particles

$$\mathbf{x}_{0:k}^{(i)} \sim q(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$$

2. Update weights

$$w_k^{(i)} \propto \frac{p(\mathbf{x}_{0:k}^{(i)}|\mathbf{y}_{1:k})}{q(\mathbf{x}_{0:k}^{(i)}|\mathbf{y}_{1:k})}$$

Comments on drawing particles:

- Let us assume that

$$q(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k)q(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}).$$

- we generate $\mathbf{x}_{0:k-1}^{(i)} \sim q(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$ at time $k-1$,
- it is sufficient to generate $\mathbf{x}_k^{(i)} \sim q(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)$ and append that to $\mathbf{x}_{1:k-1}^{(i)}$!

- It remains to derive the expression for the weights:

$$w_k^{(i)} \propto \frac{p(\mathbf{x}_{0:k}^{(i)}|\mathbf{y}_{1:k})}{q(\mathbf{x}_{0:k}^{(i)}|\mathbf{y}_{1:k})}$$

$\propto p(x_{0:k-1}^{(i)}, x_k^{(i)}, y_k | y_{1:k-1}) = p(y_k | x_k^{(i)}) p(x_k^{(i)} | x_{k-1}^{(i)}) p(x_{0:k-1}^{(i)} | y_{1:k-1})$

$$\propto \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(i)})p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)} \underbrace{\frac{p(\mathbf{x}_{0:k-1}^{(i)}|\mathbf{y}_{1:k-1})}{q(\mathbf{x}_{0:k-1}^{(i)}|\mathbf{y}_{1:k-1})}}_{w_{k-1}^{(i)}}$$

$$\propto w_{k-1}^{(i)} \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(i)})p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)}$$

- We have thus derived the SIS algorithm:

## Standard SIS algorithm

- For $i = 1, \ldots, N$ and at each time $k$:
  - Draw $\mathbf{x}_k^{(i)} \sim q(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)$.
  - Compute weights
  $$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)}.$$
  - Normalize the weights.

- A simple choice of importance density is

$$q(\mathbf{x}_k | \mathbf{x}_{k-1}, \cancel{\mathbf{y}_k}) = p(\mathbf{x}_k | \mathbf{x}_{k-1})$$

for which $w_k^{(i)} \propto w_{k-1}^{(i)} p(\mathbf{y}_k | \mathbf{x}_k^{(i)})$.

## Example - Nonlinear filter benchmark

- The following is a common benchmark for nonlinear filters

$$
\begin{aligned}
x_k &= \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) + q_{k-1} \\
y_k &= \frac{x_k^2}{20} + r_k
\end{aligned}
$$

where $q_{k-1} \sim \mathcal{N}(0, 10)$ and $r_k \sim \mathcal{N}(0, 1)$.

- Let us see how the above filter performs on this challenging problem!