Introduction to Sequential Monte Carlo Methods

Arnaud Doucet

NCSU, October 2008
**Preliminary Remarks**

- *Sequential Monte Carlo* (SMC) are a set of methods allowing us to approximate virtually *any sequence of probability distributions*.
- SMC are very popular in physics where they are used to compute eigenvalues of positive operators, the solution of PDEs/integral equations or simulate polymers.
- We focus here on *Applications of SMC to Hidden Markov Models* (HMM) for pedagogical reasons...
- In the HMM framework, SMC are also widely known as Particle Filtering/Smoothing methods.
Markov Models

- We model the stochastic processes of interest as a discrete-time Markov process \( \{X_k\}_{k \geq 1} \).
- \( \{X_k\}_{k \geq 1} \) is characterized by its *initial density* 
  \[ X_1 \sim \mu(\cdot) \]
  and its *transition density*
  \[ X_k \mid (X_{k-1} = x_{k-1}) \sim f(\cdot \mid x_{k-1}). \]
- We introduce the notation \( x_{i:j} = (x_i, x_{i+1}, \ldots, x_j) \) for \( i \leq j \). We have by definition
  \[ p(x_{1:n}) = p(x_1) \prod_{k=2}^{n} p(x_k \mid x_{1:k-1}) = \mu(x_1) \prod_{k=2}^{n} f(x_k \mid x_{k-1}) \]
Observation Model

- We do not observe \( \{X_k\}_{k \geq 1} \); the process is *hidden*. We only have access to another related process \( \{Y_k\}_{k \geq 1} \).
- We assume that, conditional on \( \{X_k\}_{k \geq 1} \), the observations \( \{Y_k\}_{k \geq 1} \) are independent and marginally distributed according to

\[
Y_k \mid (X_k = x_k) \sim g(\cdot \mid x_k).
\]

- Formally this means that

\[
p(y_{1:n} \mid x_{1:n}) = \prod_{k=1}^{n} g(y_k \mid x_k).
\]
Figure: Graphical model representation of HMM
Assume you want to track a target in the XY plane then you can consider the 4-dimensional state

\[ X_k = (X_{k,1}, V_{k,1}, X_{k,2}, V_{k,2})^T \]

The so-called constant velocity model states that

\[ X_k = AX_{k-1} + W_k, \quad W_k \sim_{\text{i.i.d.}} \mathcal{N}(0, \Sigma), \]

\[ A = \begin{pmatrix} A_{CV} & 0 \\ 0 & A_{CV} \end{pmatrix}, \quad A_{CV} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}, \]

\[ \Sigma = \sigma^2 \begin{pmatrix} \Sigma_{CV} & 0 \\ 0 & \Sigma_{CV} \end{pmatrix}, \quad \Sigma_{CV} = \begin{pmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{pmatrix} \]

We obtain that

\[ f(x_k | x_{k-1}) = \mathcal{N}(x_k; Ax_{k-1}, \Sigma). \]
The observation equation is dependent on the sensor.

**Simple case**

\[ Y_k = CX_k + DE_k, \quad E_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma_e) \]

so

\[ g(y_k | x_k) = \mathcal{N}(y_k; Cx_k, \Sigma_e). \]

**Complex realistic case (Bearings-only-tracking)**

\[ Y_k = \tan^{-1} \left( \frac{X_{k,2}}{X_{k,1}} \right) + E_k, \quad E_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2) \]

so

\[ g(y_k | x_k) = \mathcal{N}(y_k; \tan^{-1} \left( \frac{X_{k,2}}{X_{k,1}} \right), \sigma^2). \]
We have the following standard model

\[ X_k = \phi X_{k-1} + V_k, \quad V_k \sim \text{i.i.d. } \mathcal{N}(0, \sigma^2) \]

so that

\[ f(x_k | x_{k-1}) = \mathcal{N}(x_k; \phi x_{k-1}, \sigma^2). \]

We observe

\[ Y_k = \beta \exp(X_k/2) W_k, \quad W_k \sim \text{i.i.d. } \mathcal{N}(0, 1) \]

so that

\[ g(y_k | x_k) = \mathcal{N}(y_k; 0, \beta^2 \exp(x_k)). \]
Inference in HMM

- Given a realization of the observations \( Y_{1:n} = y_{1:n} \), we are interested in inferring the states \( X_{1:n} \).
- We are in a Bayesian framework where

\[
\text{Prior: } p(x_{1:n}) = \mu(x_1) \prod_{k=2}^{n} f(x_k | x_{k-1}),
\]

\[
\text{Likelihood: } p(y_{1:n} | x_{1:n}) = \prod_{k=1}^{n} g(y_k | x_k)
\]

- Using Bayes’ rule, we obtain

\[
p(x_{1:n} | y_{1:n}) = \frac{p(y_{1:n} | x_{1:n}) p(x_{1:n})}{p(y_{1:n})}
\]

where the marginal likelihood is given by

\[
p(y_{1:n}) = \int p(y_{1:n} | x_{1:n}) p(x_{1:n}) \, dx_{1:n}.
\]
In particular, we will focus here on the *sequential estimation* of 
$p(x_{1:n} \mid y_{1:n})$ and $p(y_{1:n})$; that is at each time $n$ we want update our 
knowledge of the hidden process in light of $y_n$.

There is a simple recursion relating $p(x_{1:n-1} \mid y_{1:n-1})$ to $p(x_{1:n} \mid y_{1:n})$ 
given by

$$
p(x_{1:n} \mid y_{1:n}) = p(x_{1:n-1} \mid y_{1:n-1}) \frac{f(x_n \mid x_{n-1}) g(y_n \mid x_n)}{p(y_n \mid y_{1:n-1})}
$$

where

$$
p(y_{n} \mid y_{1:n-1}) = \int g(y_n \mid x_n) f(x_n \mid x_{n-1}) p(x_{n-1} \mid y_{1:n-1}) \, dx_{n-1:n}.
$$

We will also simply write

$$
p(x_{1:n} \mid y_{1:n}) \propto p(x_{1:n-1} \mid y_{1:n-1}) f(x_n \mid x_{n-1}) g(y_n \mid x_n).
$$
In many papers/books in the literature, you will find the following two-step prediction-updating recursion for the marginals so-called filtering distributions $p(x_n|y_{1:n})$ which is a direct consequence.

**Prediction Step**

$$p(x_n|y_{1:n-1}) = \int p(x_{n-1:n}|y_{1:n-1}) \, dx_{n-1}$$

$$= \int p(x_n|x_{n-1}, y_{1:n-1}) p(x_{n-1}|y_{1:n-1}) \, dx_{n-1}$$

$$= \int \tilde{f}(x_n|x_{n-1}) p(x_{n-1}|y_{1:n-1}) \, dx_{n-1}.$$  

**Updating Step**

$$p(x_n|y_{1:n}) = \frac{g(y_n|x_n) p(x_n|y_{1:n-1})}{p(y_n|y_{1:n-1})}$$
We have seen that
\[ p(y_{1:n}) = \int p(y_{1:n} | x_{1:n}) p(x_{1:n}) \, dx_{1:n}. \]

We also have the following decomposition
\[
p(y_{1:n}) = p(y_1) \prod_{k=2}^{n} p(y_k | y_{1:k-1})
\]

where
\[
p(y_k | y_{1:k-1}) = \int p(y_k, x_k | y_{1:k-1}) \, dx_k
\]
\[= \int g(y_k | x_k) p(x_k | y_{1:k-1}) \, dx_k
\]
\[= \int g(y_k | x_k) f(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) \, dx_{k-1}
\]

We have "broken" an high dimensional integral into the product of lower dimensional integrals.
Closed-form Inference in HMM

- We have closed-form solutions for
  - Finite state-space HMM; i.e. \( E = \{ e_1, \ldots, e_p \} \) as all integrals are becoming finite sums
  - Linear Gaussian models; all the posterior distributions are Gaussian; e.g. the celebrated Kalman filter.
  - A whole reverse engineering literature exists for closed-form solutions in alternative cases...

- In many cases of interest, it is impossible to compute the solution in closed-form and we need approximations,
Gaussian approximations: Extended Kalman filter, Unscented Kalman filter.

Gaussian sum approximations.

Projection filters, Variational approximations.

Simple discretization of the state-space.

Analytical methods work in simple cases but are not reliable and it is difficult to diagnose when they fail.

Standard discretization of the space is expensive and difficult to implement in high-dimensional scenarios.
At the beginning of the 90’s, the optimal filtering area was considered virtually dead; there had not been any significant progress for years then...


This article introduces a simple method which relies neither on a functional approximation nor a deterministic grid.

This paper was ignored by most researchers for a few years...
- Monte Carlo Sampling.
- Importance Sampling.
- Sequential Importance Sampling.
- Sequential Importance Sampling with Resampling.
Monte Carlo Sampling

- Assume for the time being that you are interested in estimating the high-dimensional probability density

\[ p(x_{1:n} | y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} \propto p(x_{1:n}, y_{1:n}) \]

where \( n \) is fixed.

- A Monte Carlo approximation consists of sampling a large number \( N \) of i.i.d. random variables \( X_{1:n}^{(i)} \sim p(x_{1:n} | y_{1:n}) \) and build the following approximation

\[ \hat{p}(x_{1:n} | y_{1:n}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}(x_{1:n}) \]

where \( \delta_{a_{1:n}}(x_{1:n}) \) is the delta-Dirac mass which is such that

\[ \int_{A} \delta_{a_{1:n}}(x_{1:n}) \, dx_{1:n} = \begin{cases} 1 & \text{if } a_{1:n} \in A \subset E^n, \\ 0 & \text{otherwise}. \end{cases} \]
Issues with Standard Monte Carlo Sampling

- There are standard methods to sample from classical distributions such as Beta, Gamma, Normal, Poisson etc. We will not detail them here although we will rely on them.

- **Problem 1:** For most problems of interest, we cannot sample from $p(x_{1:n} \mid y_{1:n})$.

- **Problem 2:** Even if we could sample exactly from $p(x_{1:n} \mid y_{1:n})$, then the computational complexity of the algorithm would most likely increase with $n$: we want here an algorithm of fixed computational complexity at each time step.

- To summarize, we cannot use standard MC sampling in our case and, even if we could, this would not solve our problem...
Importance Sampling (IS). We have

\[
p(x_{1:n} | y_{1:n}) = \frac{p(y_{1:n} | x_{1:n}) p(x_{1:n})}{p(y_{1:n})},
\]

\[
p(y_{1:n}) = \int p(y_{1:n} | x_{1:n}) p(x_{1:n}) \, dx_{1:n}
\]

Generally speaking, we have for a so-called *importance distribution* \(q(x_{1:n} | y_{1:n})\) such that selected such that

\[
p(x_{1:n} | y_{1:n}) > 0 \Rightarrow q(x_{1:n} | y_{1:n}) > 0
\]

\[
p(x_{1:n} | y_{1:n}) = \frac{w(x_{1:n}, y_{1:n}) q(x_{1:n} | y_{1:n})}{p(y_{1:n})},
\]

\[
p(y_{1:n}) = \int w(x_{1:n}, y_{1:n}) q(x_{1:n} | y_{1:n}) \, dx_{1:n}
\]

where the unnormalized *importance weight* is

\[
w(x_{1:n}, y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{q(x_{1:n} | y_{1:n})} \propto \frac{p(x_{1:n} | y_{1:n})}{q(x_{1:n} | y_{1:n})}.
\]
Monte Carlo IS Estimates

- It is easy to sample from \( p(x_{1:n}) \) thus we can build the standard MC approximation

\[
\hat{p}(x_{1:n} | y_{1:n}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}(x_{1:n}) \text{ where } X_{1:n}^{(i)} \overset{\text{i.i.d.}}{\sim} p(x_{1:n}).
\]

- We plug these approximations in the IS identities to obtain

\[
p(y_{1:n}) = \int p(y_{1:n} | x_{1:n}) p(x_{1:n}) \, dx_{1:n},
\]

\[
\Rightarrow \hat{p}(y_{1:n}) = \frac{1}{N} \sum_{i=1}^{N} p \left( y_{1:n} | X_{1:n}^{(i)} \right).
\]

- \( \hat{p}(y_{1:n}) \) is an unbiased estimate of \( p(y_{1:n}) \) with variance

\[
\frac{1}{N} \left[ \int p^2(y_{1:n} | x_{1:n}) p(x_{1:n}) \, dx_{1:n} - 1 \right].
\]
We also get an approximation of the posterior

\[
p ( x_{1:n} | y_{1:n} ) = \frac{p ( y_{1:n} | x_{1:n} ) p ( x_{1:n} )}{\int p ( y_{1:n} | x_{1:n} ) p ( x_{1:n} ) \, dx_{1:n}}
\]

using

\[
\hat{p} ( x_{1:n} | y_{1:n} ) = \frac{p ( y_{1:n} | x_{1:n} ) \hat{p} ( x_{1:n} )}{\int p ( y_{1:n} | x_{1:n} ) \hat{p} ( x_{1:n} ) \, dx_{1:n}}
\]

\[
= \frac{1}{N} \sum_{i=1}^{N} p ( y_{1:n} | X_{1:n}^{(i)} ) \delta_{X_{1:n}^{(i)}} ( x_{1:n} )
\]

\[
= \sum_{i=1}^{N} \mathcal{W}_{n}^{(i)} \delta_{X_{1:n}^{(i)}} ( x_{1:n} )
\]

where the \textit{normalized importance weights} are

\[
\mathcal{W}_{n}^{(i)} = \frac{p ( y_{1:n} | X_{1:n}^{(i)} )}{\sum_{j=1}^{N} p ( y_{1:n} | X_{1:n}^{(j)} )}.
\]
Assume we are interested in computing $\mathbb{E}_{p(x_{1:n}|y_{1:n})}(\varphi)$, then we can use the estimate

$$\mathbb{E}_{\hat{p}(x_{1:n}|y_{1:n})}(\varphi) = \sum_{i=1}^{N} W_{n}^{(i)} \varphi \left( X_{1:n}^{(i)} \right).$$

This estimate is biased for a finite $N$ but is asymptotically consistent with

$$\lim_{N \to \infty} N \left( \mathbb{E}_{\hat{p}(x_{1:n}|y_{1:n})}(\varphi) - \mathbb{E}_{p(x_{1:n}|y_{1:n})}(\varphi) \right) = - \int \frac{p^2(x_{1:n}|y_{1:n})}{p(x_{1:n})} \left( \varphi(x_{1:n}) - \mathbb{E}_{p(x_{1:n}|y_{1:n})}(\varphi) \right) dx_{1:n}$$

and

$$\sqrt{N} \left( \mathbb{E}_{\hat{p}(x_{1:n}|y_{1:n})}(\varphi) - \mathbb{E}_{p(x_{1:n}|y_{1:n})}(\varphi) \right) \Rightarrow \mathcal{N}(0, \int \frac{p^2(x_{1:n}|y_{1:n})}{p(x_{1:n})} \left( \varphi(x_{1:n}) - \mathbb{E}_{p(x_{1:n}|y_{1:n})}(\varphi) \right)^2 dx_{1:n}).$$

$$MSE = \frac{bias^2}{O(N^{-2})} + \frac{variance}{O(N^{-1})}$$

so asymptotic bias is irrelevant.
Summary of Our Progresses

- **Problem 1**: For most problems of interest, we cannot sample from $p(x_{1:n} | y_{1:n})$.

- **Problem 1 “solved”**: We use an IS approximation of $p(x_{1:n} | y_{1:n})$ that relies on the IS prior distribution $p(x_{1:n})$.

- **Problem 2**: Even if we could sample exactly from $p(x_{1:n} | y_{1:n})$, then the computational complexity of the algorithm would most likely increase with $n$: we want here an algorithm of fixed computational complexity at each time step.

- **Problem 2 not solved yet**: If at each time step $n$, we need to obtain new samples from $p(x_{1:n})$ then the algorithm computational complexity will increase at each time step.
Sequential Importance Sampling (SIS)

To avoid having computational efforts increasing over time, we use the fact that

\[
\frac{p(x_{1:n})}{p(x_{1:n-1}) \times f(x_n|x_{n-1})} = \mu(x_1) \prod_{k=2}^{n} f(x_k|x_{k-1}).
\]

In practical terms, this means that at time \( n - 1 \), we have already sampled \( X_{1:n-1} \) and that to obtain at time \( n \) samples/particles \( X_{1:n} \), we just need to sample

\[
X_{n}^{(i)} \mid X_{n-1}^{(i)} \sim f(x_n|X_{n-1}^{(i)})
\]

and set

\[
X_{1:n}^{(i)} = (X_{1:n-1}^{(i)}, X_{n}^{(i)})
\]

previously sampled paths, new sampled component.
Now, whatever being \( n \), we have only one component \( X_n \) to sample!

However, can we compute our IS estimates of \( p (y_{1:n}) \) and the target \( p (x_{1:n} | y_{1:n}) \) recursively?

Remember that

\[
\hat{p} (y_{1:n}) = \frac{1}{N} \sum_{i=1}^{N} p \left( y_{1:n} | X_{1:n}^{(i)} \right),
\]

\[
\hat{p} (x_{1:n} | y_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_{1:n}^{(i)}} (x_{1:n}),
\]

where \( W_n^{(i)} \propto p \left( y_{1:n} | X_{1:n}^{(i)} \right), \sum_{i=1}^{N} W_n^{(i)} = 1. \)

We have

\[
p \left( y_{1:n} | x_{1:n} \right) = p \left( y_{1:n-1} | x_{1:n-1} \right) g \left( y_n | x_n \right)
\]
Sequential Importance Sampling Algorithm

- **At time 1,**
  Sample $N$ particles $X_1^{(i)} \sim \mu(x_1)$ and compute
  
  $$W_1^{(i)} \propto g(y_1 | X_1^{(i)}).$$

- **At time $n$, $n \geq 2$**
  Sample $N$ particles $X_n^{(i)} \sim f(x_n | X_{n-1}^{(i)})$ and compute
  
  $$W_n^{(i)} \propto W_{n-1}^{(i)} \cdot g(y_n | X_n^{(i)}).$$
The algorithm can be easily parallelized.

The computational complexity does not increase over time.

It is not necessary to store the paths \( \{ X_{1:n}^{(i)} \} \) if we are only interested in approximating \( p(x_n | y_{1:n}) \) as the weights only depends on \( \{ X_{n-1:n}^{(i)} \} \)!
Consider the following model

\[ X_k = 0.5X_{k-1} + \frac{25X_{k-1}}{1 + X^2_{k-1}} + 8\cos(1.2k) + V_k \]

\[ = \varphi(X_{k-1}) + V_k \]

\[ Y_k = \frac{X^2_k}{20} + W_k, \]

where \( X_1 \sim \mathcal{N}(0, 1) \), \( V_k \sim \text{i.i.d. } \mathcal{N}(0, 2.5^2) \) and \( W_k \sim \text{i.i.d. } \mathcal{N}(0, 1) \).
Figure: Histogram of $\log \left( p \left( y_{1:100} | X_{1:100}^{(i)} \right) \right)$. The approximation is dominated by one single particle.
SIS is an attractive idea: *sequential* and *parallelizable*, reduces the design of an high-dimensional proposal to the design of a sequence of low-dimensional proposals.

SIS can only work for moderate size problems.

Is there a way to *partially* fix this problem?
Resampling

- **Problem**: As $n$ increases, the variance of $\{ p \left( y_{1:n} \mid X_{1:n}^{(i)} \right) \}$ increases and all the mass is concentrated on a few random samples/particles.

\[
\hat{p} \left( x_{1:n} \mid y_{1:n} \right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_{1:n}^{(i)}} (x_{1:n}) \approx \delta_{X_{1:n}^{(i_0)}} (x_{1:n})
\]

as $W_n^{(i_0)} \approx 1$ and $W_n^{(i)} \approx 0$ for $i \neq i_0$.

- **Intuitive KEY idea**: Kill in a principled way the particles with low weights $W_n^{(i)}$ (relative to $1/N$) and multiply the particles with high weights $W_n^{(i)}$ (relative to $1/N$).

- **Rationale**: If a particle at time $n$ has a low weight then typically it will still have a low weight at time $n+1$ (though I can easily give you a counterexample) and you want to focus your computational efforts on the “promising” parts of the space.
At time $n$, IS provides the following approximation of $p(x_{1:n} | y_{1:n})$

$$\hat{p}(x_{1:n} | y_{1:n}) = \sum_{i=1}^{N} W_n(i) \delta_{\chi_{1:n}^{(i)}}(x_{1:n}) .$$

The simplest resampling schemes consists of sampling $N$ times $\tilde{X}_{1:n}^{(i)} \sim \hat{p}(x_{1:n} | y_{1:n})$ to build the new approximation

$$\tilde{p}(x_{1:n} | y_{1:n}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\tilde{X}_{1:n}^{(i)}}(x_{1:n}) .$$

The new resampled particles $\{\tilde{X}_{1:n}^{(i)}\}$ are approximately distributed according to $p(x_{1:n} | y_{1:n})$ but statistically dependent. Theoretically much more difficult to study.
Sequential Importance Sampling Resampling Algorithm

- **At time 1,**
  Sample $N$ particles $X_{1}^{(i)} \sim \mu(x_1)$ and compute
  
  $$W_{1}^{(i)} \propto g(y_1 | X_{1}^{(i)}).$$

  Resample $\{X_{1}^{(i)}, W_{1}^{(i)}\}$ to obtain new particles also denoted $\{X_{1}^{(i)}\}$.

- **At time n, $n \geq 2$**
  Sample $N$ particles $X_{n}^{(i)} \sim f(x_n | X_{n-1}^{(i)})$ and compute
  
  $$W_{n}^{(i)} \propto g(y_n | X_{n}^{(i)}).$$

  Resample $\{X_{1:n}^{(i)}, W_{n}^{(i)}\}$ to obtain new particles also denoted $\{X_{1:n}^{(i)}\}$.
We also have

\[ p(y_n | y_{1:n-1}) = \int g(y_n | x_n) f(x_n | x_{n-1}) p(x_{n-1} | y_{1:n-1}) \, dx_{n-1:n} \]

so

\[ \hat{p}(y_n | y_{1:n-1}) = \frac{1}{N} \sum_{i=1}^{N} g(y_n | x^{(i)}_n) . \]

Perhaps surprisingly, it can be shown that if we define

\[ \hat{p}(y_{1:n}) = \hat{p}(y_1) \prod_{k=2}^{n} \hat{p}(y_k | y_{1:k-1}) \]

then

\[ \mathbb{E}[\hat{p}(y_{1:n})] = p(y_{1:n}) . \]
Consider again the following model

\[ X_k = 0.5X_{k-1} + \frac{25X_{k-1}}{1 + X_{k-1}^2} + 8\cos(1.2k) + V_k \]

\[ Y_k = \frac{X_k^2}{20} + W_k, \]

where \( X_1 \sim \mathcal{N}(0, 1) \), \( V_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 2.5^2) \) and \( W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1) \).
I have presented the most basic algorithm.

In practice, practitioners often select an IS distribution
\[ q(x_n \mid y_n, x_{n-1}) \neq f(x_n \mid x_{n-1}) \]. In such cases, we have

\[ W_n(i) \propto \frac{f(X_n(i) \mid X_{n-1}(i))}{q(X_n(i) \mid y_n, X_{n-1}(i))} g(y_n \mid X_n(i)) \]

Better resampling steps have been developed.

Variance reduction can also be developed.

SMC methods can be used to sample from virtually any sequence of distributions.