# 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*

$$X_k | (X_{k-1} = x_{k-1}) \sim f(\cdot | x_{k-1}).$$

- We introduce the notation $x_{i:j} = (x_i, x_{i+1}, ..., x_j)$ for $i \leq j$. We have by definition

$$\begin{aligned} p(x_{1:n}) &= p(x_1) \prod_{k=2}^{n} p(x_k | x_{1:k-1}) \\ &= \mu(x_1) \prod_{k=2}^{n} f(x_k | x_{k-1}) \end{aligned}$$

# 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

$$Y_k | (X_k = x_k) \sim g(\cdot | x_k).$$

- Formally this means that

$$p(y_{1:n} | x_{1:n}) = \prod_{k=1}^{n} g(y_k | x_k).$$

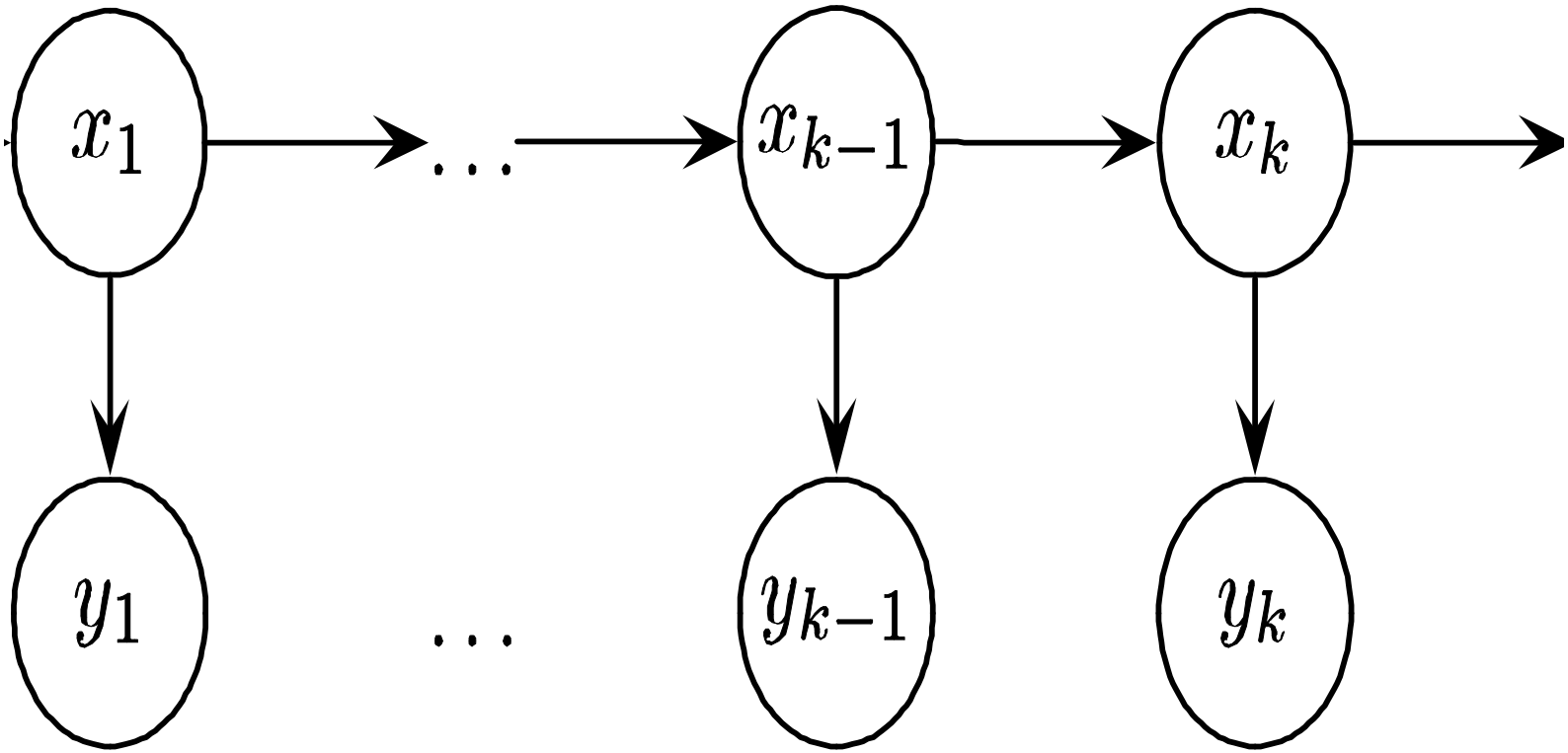


Figure: Graphical model representation of HMM

# Tracking Example

- Assume you want to track a target in the XY plane then you can consider the 4-dimensional state

$$X_k = (X_{k,1}, V_{k,1}, X_{k,2}, V_{k,2})^{\mathrm{T}}$$

- The so-called constant velocity model states that

$$X_k = AX_{k-1} + W_k, \; W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma),$$

$$A = \begin{pmatrix} A_{CV} & 0 \\ 0 & A_{CV} \end{pmatrix}, A_{CV} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix},$$

$$\Sigma = \sigma^2 \begin{pmatrix} \Sigma_{CV} & 0 \\ 0 & \Sigma_{CV} \end{pmatrix}, \; \Sigma_{CV} = \begin{pmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{pmatrix}$$

- We obtain that

$$f(x_k | x_{k-1}) = \mathcal{N}(x_k; Ax_{k-1}, \Sigma).$$

# Tracking Example (cont.)

- The observation equation is dependent on the sensor.
- *Simple case*

$$Y_k = CX_k + DE_k, \;\; E_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma_e)$$

so

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

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

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

so

$$g(y_k | x_k) = \mathcal{N}\left(y_k; \tan^{-1}\left(\frac{x_{k,2}}{x_{k,1}}\right), \sigma^2\right).$$

# Stochastic Volatility

- We have the following standard model

$$X_k = \phi X_{k-1} + V_k, \;\; V_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma^2\right)$$

so that

$$f\left(x_k \middle| x_{k-1}\right) = \mathcal{N}\left(x_k; \phi x_{k-1}, \sigma^2\right).$$

- We observe

$$Y_k = \beta \exp\left(X_k/2\right) W_k, \;\; W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, 1\right)$$

so that

$$g\left(y_k \middle| x_k\right) = \mathcal{N}\left(y_k; 0, \beta^2 \exp\left(x_k\right)\right).$$

# 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

$$\textit{Prior}: \; p\left(x_{1:n}\right) = \mu\left(x_1\right) \prod_{k=2}^{n} f\left(x_k \middle| x_{k-1}\right),$$

$$\textit{Likelihood}: \; p\left(y_{1:n} \middle| x_{1:n}\right) = \prod_{k=1}^{n} g\left(y_k \middle| x_k\right)$$

- Using Bayes' rule, we obtain

$$p\left(x_{1:n} \middle| y_{1:n}\right) = \frac{p\left(y_{1:n} \middle| x_{1:n}\right) p\left(x_{1:n}\right)}{p\left(y_{1:n}\right)}$$

where the marginal likelihood is given by

$$p\left(y_{1:n}\right) = \int p\left(y_{1:n} \middle| x_{1:n}\right) p\left(x_{1:n}\right) dx_{1:n}.$$

# Sequential Inference in HMM

- In particular, we will focus here on the *sequential estimation* of $p\left(x_{1:n}\middle|y_{1:n}\right)$ and $p\left(y_{1:n}\right)$; 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 $p\left(x_{1:n-1}\middle|y_{1:n-1}\right)$ to $p\left(x_{1:n}\middle|y_{1:n}\right)$ given by

$$p\left(x_{1:n}\middle|y_{1:n}\right)=p\left(x_{1:n-1}\middle|y_{1:n-1}\right)\frac{f\left(x_n\middle|x_{n-1}\right)g\left(y_n\middle|x_n\right)}{p\left(y_n\middle|y_{1:n-1}\right)}$$

where

$$p\left(y_n\middle|y_{1:n-1}\right)=\int g\left(y_n\middle|x_n\right)f\left(x_n\middle|x_{n-1}\right)p\left(x_{n-1}\middle|y_{1:n-1}\right)dx_{n-1:n}.$$

- We will also simply write

$$p\left(x_{1:n}\middle|y_{1:n}\right)\propto p\left(x_{1:n-1}\middle|y_{1:n-1}\right)f\left(x_n\middle|x_{n-1}\right)g\left(y_n\middle|x_n\right).$$

- 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*

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

- *Updating Step*

$$p(x_n | y_{1:n}) = \frac{g(y_n | x_n) \, p(x_n | y_{1:n-1})}{p(y_n | y_{1:n-1})}$$

# (Marginal) Likelihood Evaluation

- We have seen that

$$p\left(y_{1:n}\right) = \int p\left(y_{1:n} \middle| x_{1:n}\right) p\left(x_{1:n}\right) dx_{1:n}.$$

- We also have the following decomposition

$$p\left(y_{1:n}\right) = p\left(y_1\right) \prod_{k=2}^{n} p\left(y_k \middle| y_{1:k-1}\right)$$

where

$$\begin{aligned} p\left(y_k \middle| y_{1:k-1}\right) &= \int p\left(y_k, x_k \middle| y_{1:k-1}\right) dx_k \\ &= \int g\left(y_k \middle| x_k\right) p\left(x_k \middle| y_{1:k-1}\right) dx_k \\ &= \int g\left(y_k \middle| x_k\right) f\left(x_k \middle| x_{k-1}\right) p\left(x_{k-1} \middle| y_{1:k-1}\right) dx_{k-1} \end{aligned}$$

- 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. $E = \{e_1, ..., e_p\}$ 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,

# Standard Approximations for Filtering Distributions

- 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.

# Breakthrough

- 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...
- Gordon, N.J. Salmond, D.J. Smith, A.F.M. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *IEE Proceedings F: Radar and Signal Processing*, vol. 140, no. 2, pp. 107-113, 1993.
- 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

$$p\left(x_{1:n}\middle|\, y_{1:n}\right) = \frac{p\left(x_{1:n}, y_{1:n}\right)}{p\left(y_{1:n}\right)} \propto p\left(x_{1:n}, y_{1:n}\right)$$

  where $n$ is *fixed*.
- A Monte Carlo approximation consists of sampling a large number $N$ of i.i.d. random variables $X_{1:n}^{(i)} \overset{\text{i.i.d.}}{\sim} p\left(x_{1:n}\middle|\, y_{1:n}\right)$ and build the following approximation

$$\widehat{p}\left(x_{1:n}\middle|\, y_{1:n}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right)$$

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

$$\int_A \delta_{a_{1:n}}\left(x_{1:n}\right) dx_{1:n} = \begin{cases} 1 & \text{if } a_{1:n} \in A \subset E^n, \\ 0 & \text{otherwise.} \end{cases}$$

# 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 $p\left(x_{1:n} \middle| y_{1:n}\right)$.
- **Problem 2**: Even if we could sample exactly from $p\left(x_{1:n} \middle| y_{1:n}\right)$, 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

- **Importance Sampling** (IS). We have

$$p\left(x_{1:n}\middle|y_{1:n}\right) = \frac{p\left(y_{1:n}\middle|x_{1:n}\right)p\left(x_{1:n}\right)}{p\left(y_{1:n}\right)},$$

$$p\left(y_{1:n}\right) = \int p\left(y_{1:n}\middle|x_{1:n}\right)p\left(x_{1:n}\right)dx_{1:n}$$

- Generally speaking, we have for a so-called *importance distribution* $q\left(x_{1:n}\middle|y_{1:n}\right)$ such that selected such that $p\left(x_{1:n}\middle|y_{1:n}\right) > 0 \Rightarrow q\left(x_{1:n}\middle|y_{1:n}\right) > 0$

$$p\left(x_{1:n}\middle|y_{1:n}\right) = \frac{w\left(x_{1:n},y_{1:n}\right)q\left(x_{1:n}\middle|y_{1:n}\right)}{p\left(y_{1:n}\right)},$$

$$p\left(y_{1:n}\right) = \int w\left(x_{1:n},y_{1:n}\right)q\left(x_{1:n}\middle|y_{1:n}\right)dx_{1:n}$$

where the unnormalized *importance weight* is

$$w\left(x_{1:n},y_{1:n}\right) = \frac{p\left(x_{1:n},y_{1:n}\right)}{q\left(x_{1:n}\middle|y_{1:n}\right)} \propto \frac{p\left(x_{1:n}\middle|y_{1:n}\right)}{q\left(x_{1:n}\middle|y_{1:n}\right)}.$$

# Monte Carlo IS Estimates

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

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

- We plug these approximations in the IS identities to obtain

$$\begin{aligned} p(y_{1:n}) &= \int p(y_{1:n}|x_{1:n})\, p(x_{1:n})\, dx_{1:n}, \\ \Rightarrow \quad \widehat{p}(y_{1:n}) &= \frac{1}{N}\sum_{i=1}^{N} p\left(y_{1:n}|X_{1:n}^{(i)}\right). \end{aligned}$$

- $\widehat{p}(y_{1:n})$ 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

$$p\left(x_{1:n}\middle|y_{1:n}\right)=\frac{p\left(y_{1:n}\middle|x_{1:n}\right)p\left(x_{1:n}\right)}{\int p\left(y_{1:n}\middle|x_{1:n}\right)p\left(x_{1:n}\right)dx_{1:n}}$$

using

$$\begin{aligned}\widehat{p}\left(x_{1:n}\middle|y_{1:n}\right) &= \frac{p\left(y_{1:n}\middle|x_{1:n}\right)\widehat{p}\left(x_{1:n}\right)}{\int p\left(y_{1:n}\middle|x_{1:n}\right)\widehat{p}\left(x_{1:n}\right)dx_{1:n}}\\ &= \frac{\frac{1}{N}\sum_{i=1}^{N}p\left(y_{1:n}\middle|X_{1:n}^{(i)}\right)\delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right)}{\frac{1}{N}\sum_{i=1}^{N}p\left(y_{1:n}\middle|X_{1:n}^{(i)}\right)}\\ &= \sum_{i=1}^{N}W_n^{(i)}\delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right)\end{aligned}$$

where the *normalized importance weights* are

$$W_n^{(i)}=\frac{p\left(y_{1:n}\middle|X_{1:n}^{(i)}\right)}{\sum_{j=1}^{N}p\left(y_{1:n}\middle|X_{1:n}^{(j)}\right)}.$$

- Assume we are interested in computing $\mathbb{E}_{p(x_{1:n}|y_{1:n})}(\varphi)$, then we can use the estimate

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

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

$$\lim_{N\to\infty} N\left(\mathbb{E}_{\widehat{p}(x_{1:n}|y_{1:n})}(\varphi) - \mathbb{E}_{p(x_{1:n}|y_{1:n})}(\varphi)\right)$$

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

and

$$\sqrt{N}\left(\mathbb{E}_{\widehat{p}(x_{1:n}|y_{1:n})}(\varphi) - \mathbb{E}_{p(x_{1:n}|y_{1:n})}(\varphi)\right)$$

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

- $MSE = \underbrace{bias^2}_{O(N^{-2})} + \underbrace{variance}_{O(N^{-1})}$ so asymptotic bias is irrelevant.

# Summary of Our Progresses

- **Problem 1**: For most problems of interest, we cannot sample from $p\left(x_{1:n} \middle| y_{1:n}\right)$.
- **Problem 1 "solved":** We use an IS approximation of $p\left(x_{1:n} \middle| y_{1:n}\right)$ that relies on the IS prior distribution $p\left(x_{1:n}\right)$.
- **Problem 2**: Even if we could sample exactly from $p\left(x_{1:n} \middle| y_{1:n}\right)$, 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 $p\left(x_{1:n}\right)$ 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

$$\underbrace{p\left(x_{1:n}\right)}_{\text{IS at time } n} = \underbrace{p\left(x_{1:n-1}\right)}_{\text{IS at time } n-1} \times \underbrace{f\left(x_n \middle| x_{n-1}\right)}_{\text{New sampled component}}$$

$$= \mu\left(x_1\right) \prod_{k=2}^{n} f\left(x_k \middle| x_{k-1}\right).$$

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

$$X_n^{(i)} \middle| X_{n-1}^{(i)} \sim f\left(x_n \middle| X_{n-1}^{(i)}\right)$$

and set

$$X_{1:n}^{(i)} = \Big( \underbrace{X_{1:n-1}^{(i)}}_{\text{previously sampled paths}} , \underbrace{X_n^{(i)}}_{\text{new sampled component}} \Big)$$

- 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

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

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

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

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

# Sequential Importance Sampling Algorithm

- *At time 1*,
  Sample $N$ particles $X_1^{(i)} \sim \mu(x_1)$ and compute

$$W_1^{(i)} \propto g\left(y_1 | X_1^{(i)}\right).$$

- *At time $n$, $n \geq 2$*
  Sample $N$ particles $X_n^{(i)} \sim f\left(x_n | X_{n-1}^{(i)}\right)$ and compute

$$W_n^{(i)} \propto W_{n-1}^{(i)} . g\left(y_n | X_n^{(i)}\right).$$

# Practical Issues

- The algorithm can be easily parallelized.
- The computational complexity does not increase over time.
- It is not necessary to store the paths $\left\{X_{1:n}^{(i)}\right\}$ if we are only interested in approximating $p\left(x_n \middle| y_{1:n}\right)$ as the weights only depends on $\left\{X_{n-1:n}^{(i)}\right\}$!

# Example of Applications

- Consider the following model

$$\begin{aligned} X_k &= 0.5X_{k-1} + \frac{25X_{k-1}}{1+X_{k-1}^2} + 8\cos(1.2k) + V_k \\ &= \varphi(X_{k-1}) + V_k \\ Y_k &= \frac{X_k^2}{20} + W_k, \end{aligned}$$

where $X_1 \sim \mathcal{N}(0,1)$, $V_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 2.5^2)$ and $W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$.

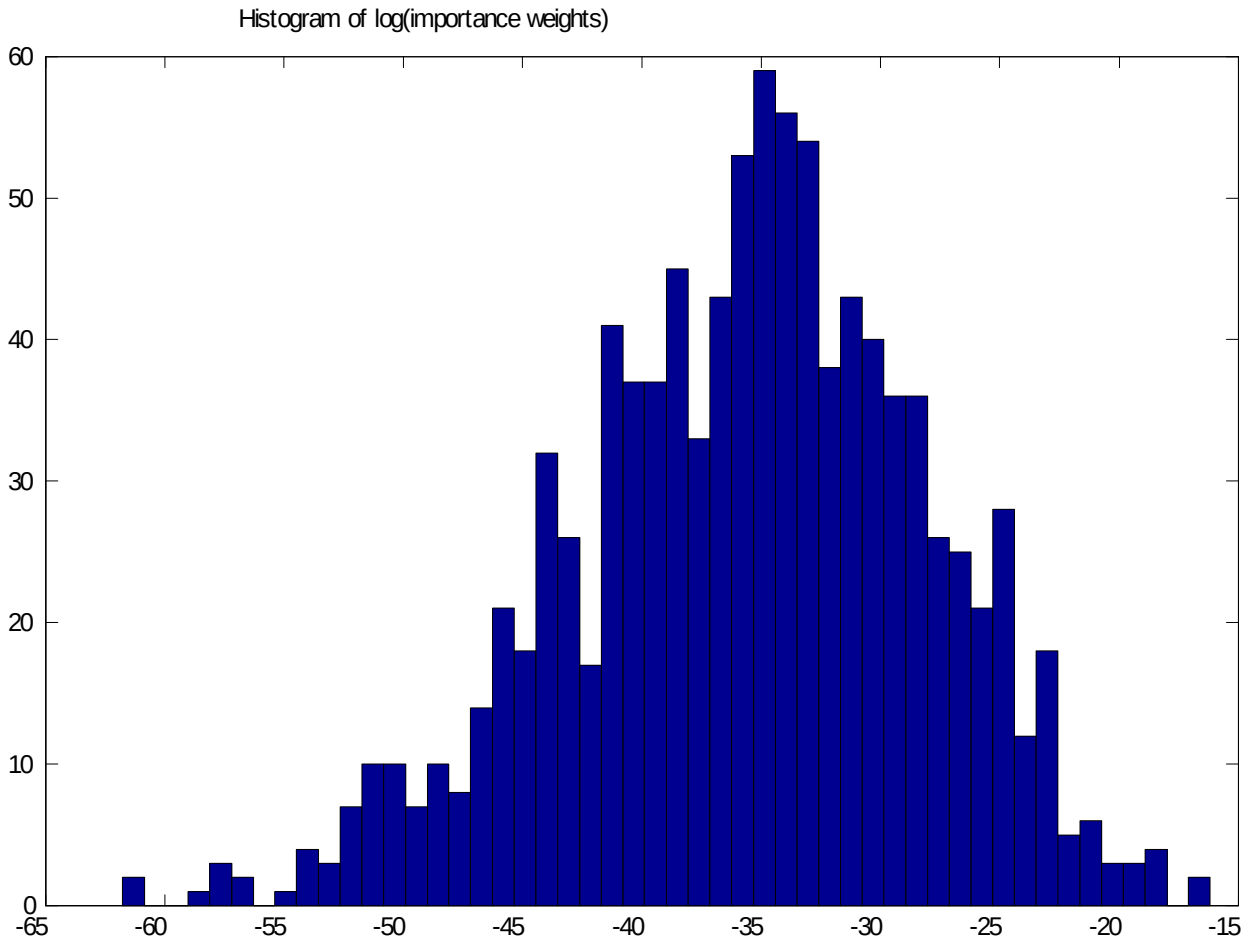


Figure: Histogram of $\log\left(p\left(y_{1:100}|X_{1:100}^{(i)}\right)\right)$. The approximation is dominated by one single particle.

# Summary

- 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 $\left\{ p\left( y_{1:n} | X_{1:n}^{(i)} \right) \right\}$ increases and all the mass is concentrated on a few random samples/particles

$$\widehat{p}\left( x_{1:n} | y_{1:n} \right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_{1:n}^{(i)}}\left( x_{1:n} \right) \approx \delta_{X_{1:n}^{(i_0)}}\left( x_{1:n} \right)$$

  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 $p\left(x_{1:n}|\, y_{1:n}\right)$

$$\widehat{p}\left(x_{1:n}|\, y_{1:n}\right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right).$$

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

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

- The new resampled particles $\left\{\widetilde{X}_{1:n}^{(i)}\right\}$ are approximately distributed according to $p\left(x_{1:n}|\, y_{1:n}\right)$ but statistically dependent $\rightsquigarrow$ 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

$$W_1^{(i)} \propto g\left(y_1 | X_1^{(i)}\right).$$

  Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$.
- *At time $n$, $n \geq 2$*
  Sample $N$ particles $X_n^{(i)} \sim f\left(x_n | X_{n-1}^{(i)}\right)$ and compute

$$W_n^{(i)} \propto g\left(y_n | X_n^{(i)}\right).$$

  Resample $\left\{X_{1:n}^{(i)}, W_n^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_{1:n}^{(i)}\right\}$.

- We also have

$$p\left(y_n \middle| y_{1:n-1}\right) = \int g\left(y_n \middle| x_n\right) f\left(x_n \middle| x_{n-1}\right) p\left(x_{n-1} \middle| y_{1:n-1}\right) dx_{n-1:n}$$

so

$$\widehat{p}\left(y_n \middle| y_{1:n-1}\right) = \frac{1}{N}\sum_{i=1}^{N} g\left(y_n \middle| X_n^{(i)}\right).$$

- Perhaps surprisingly, it can be shown that if we define

$$\widehat{p}\left(y_{1:n}\right) = \widehat{p}\left(y_1\right)\prod_{k=2}^{n}\widehat{p}\left(y_k \middle| y_{1:k-1}\right)$$

then

$$\mathbb{E}\left[\widehat{p}\left(y_{1:n}\right)\right] = p\left(y_{1:n}\right).$$

# Example (cont.)

- Consider again the following model

$$\begin{aligned} X_k &= 0.5X_{k-1} + \frac{25X_{k-1}}{1+X_{k-1}^2} + 8\cos(1.2k) + V_k \\ Y_k &= \frac{X_k^2}{20} + W_k, \end{aligned}$$

where $X_1 \sim \mathcal{N}(0,1)$, $V_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 2.5^2)$ and $W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$.

# Advanced SMC Methods

- I have presented the most basic algorithm.
- In practice, practitioners often select an IS distribution $q\left(x_n \middle| y_n, x_{n-1}\right) \neq f\left(x_n \middle| x_{n-1}\right)$. In such cases, we have

$$W_n^{(i)} \propto \frac{f\left(X_n^{(i)} \middle| X_{n-1}^{(i)}\right) g\left(y_n \middle| X_n^{(i)}\right)}{q\left(X_n^{(i)} \middle| y_n, X_{n-1}^{(i)}\right)}$$

- 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.