Stat 535 C - Statistical Computing & Monte Carlo Methods

Lecture 20 - 23rd March 2006

Arnaud Doucet

Email: arnaud@cs.ubc.ca

1.1– Outline

- Nonlinear Non Gaussian Dynamic Models.
- Sequential Bayesian Inference.
- Sequential Importance Sampling.
- Sequential Importance Sampling Resampling.

2.1– What have we done so far?

- MCMC are iterative algorithms to sample from a fixed target distribution $\pi(x) \propto f(x)$ defined on $\mathcal{X}$.
- MCMC methods can also be used to estimate the normalizing constant $\int f(x)\,dx$ although there is no simple efficient method.
- MCMC methods are not adapted to sequential Bayesian inference where the posterior has to be recomputed each time a new observation is received.
- Generally speaking MCMC are not useful when the target distribution is "time-varying"; annealing is an exception but it requires target variations to decrease over time.

2.1– What have we done so far?

- Today we will present an alternative set of methods which allows us to estimate "time-varying" targets.
- These methods are non-iterative methods and rely on Importance sampling and resampling mechanisms.
- For sake of illustration, we will detail here an application to nonlinear non-Gaussian state-space models.
- We will show in the next lectures that the methodology is much more general.

3.2– Some Examples

• *Stochastic Volatility model*

$$X_k = \phi X_{k-1} + \sigma V_k,\ V_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1),\ X_1 \sim \mathcal{N}\left(0, \frac{\sigma}{1-\phi^2}\right)$$

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

• *Bearings only tracking*

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

$$Y_k = \tan^{-1}\left(\frac{X_{k,3}}{X_{k,1}}\right) + \sigma W_k,\ W_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$

• In both cases, we have typically high-frequency data.

3.1– Nonlinear non-Gaussian State-space models

• Remember that this class of models is extremely general and includes for example

$$X_k = \varphi(X_{k-1}, V_k) \text{ where } V_k \overset{\text{i.i.d.}}{\sim} f_V,$$

$$Y_k = \psi(X_k, W_k) \text{ where } W_k \overset{\text{i.i.d.}}{\sim} g_V.$$

• See Lecture 12 for numerous examples.

3.1– Nonlinear non-Gaussian State-space models

3.1– Nonlinear non-Gaussian State-space models

• A nonlinear non-Gaussian state-space model is defined by a pair of stochastic processes $\{X_k\}_{k\geq 1}$ and $\{Y_k\}_{k\geq 1}$. $\{X_k\}_{k\geq 1}$ is an unobserved (hidden) Markov process defined by

$$X_1 \sim \mu,\ \ X_k | (X_{k-1} = x_{k-1}) \sim f(\cdot | x_{k-1}).$$

The observations $\{Y_k\}_{k\geq 1}$ are conditionally independent given $\{X_k\}_{k\geq 1}$ and

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

• The aim is to recover optimally (in a sense to precise) $\{X_k\}_{k\geq 1}$ given $\{Y_k\}_{k\geq 1}$.

4.4– Recursion for the successive target distributions

• The basic idea consists of reusing the approximation of $p\left(x_{1:n}\middle|\, y_{1:n}\right)$ available at time n to generate an approximation of $p\left(x_{1:n+1}\middle|\, y_{1:n+1}\right)$.

• One has

$$\begin{aligned} p\left(x_{1:n+1}\middle|\, y_{1:n+1}\right) &= \frac{p\left(y_{n+1}\middle|\, x_{1:n+1}, y_{1:n}\right) p\left(x_{1:n+1}\middle|\, y_{1:n}\right)}{p\left(y_{n+1}\middle|\, y_{1:n}\right)} \\ &= \frac{p\left(y_{n+1}\middle|\, x_{1:n+1}, y_{1:n}\right) p\left(x_{n+1}\middle|\, x_{1:n}, y_{1:n}\right) p\left(x_{1:n}\middle|\, y_{1:n}\right)}{p\left(y_{n+1}\middle|\, y_{1:n}\right)} \\ &= \frac{g\left(y_{n+1}\middle|\, x_{n+1}\right) f\left(x_{n+1}\middle|\, x_{n}\right) p\left(x_{1:n}\middle|\, y_{1:n}\right)}{p\left(y_{n+1}\middle|\, y_{1:n}\right)}. \end{aligned}$$

4.3– Bayesian Computation

• We have seen before how to estimate $p\left(x_{1:n}\middle|\, y_{1:n}\right)$ using MCMC.

• However, in many real-world applications, each time we receive a new observation say y_{n+1} at time $n+1$, we want to update our knowledge, that is compute $p\left(x_{1:n+1}\middle|\, y_{1:n+1}\right)$ and in particular we are often interested in $p\left(x_{n+1}\middle|\, y_{1:n+1}\right)$.

• We could run a new MCMC of invariant $p\left(x_{1:n+1}\middle|\, y_{1:n+1}\right)$ but this is computationally expensive and the computational complexity would increase over time!

• We would like to have an algorithm whose computational complexity is independent of the time index n.

4.2– Bayesian Inference

• Inference about $X_{1:n}$ given a realization of the observations $y_{1:n}$ is based on

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

• We might also be interested in computing the marginal likelihood for model choice or ML parameter estimation

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

• Typically this posterior and the marginal likelihood does not admit a closed-form expression except in the (very) important cases where $\{X_k\}$ takes values in a finite state-space or $\{X_k\}$ & $\{Y_k\}$ follow linear Gaussian equations.

4.1– Bayesian model

• The evolution equation defines a prior for $X_{1:n} = (X_1, ..., X_n)$

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

• The observation equation defines a likelihood

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

• We are “naturally” in a Bayesian framework.

4.5– Building Sequentially Monte Carlo Approximations

- We sample N particles (random samples)

$$X_1^{(i)} \sim q_1\left(x_1 \middle| y_1\right)$$

and obtain the approximation

$$p^N\left(x_1 \middle| y_1\right) = \sum_{i=1}^{N} W_1^{(i)} \delta_{X_1^{(i)}}\left(x_1\right)$$

where

$$W_1^{(i)} = \frac{w_1\left(X_1^{(i)}, y_1\right)}{\sum_{j=1}^{N} w_1\left(X_1^{(j)}, y_1\right)}.$$

4.5– Building Sequentially Monte Carlo Approximations

- Assume that you are at time 1 and want to approximate $p\left(x_1 \middle| y_1\right)$ then, because the state is usually of reasonable dimension, you can use importance sampling.

- We select an importance distribution $q_1\left(x_1 \middle| y_1\right)$ and use the identity

$$p\left(x_1 \middle| y_1\right) = \frac{w_1\left(x_1, y_1\right) q_1\left(x_1 \middle| y_1\right)}{\int w_1\left(x_1, y_1\right) q_1\left(x_1 \middle| y_1\right) dx_1}$$

where

$$w_1\left(x_1, y_1\right) = \frac{p\left(x_1, y_1\right)}{q_1\left(x_1 \middle| y_1\right)}.$$

4.4– Recursion for the successive target distributions

- In most of the literature, you'll find the following recursion on the *marginal* distributions $\left\{p\left(x_n \middle| y_{1:n}\right)\right\}$

$$\begin{aligned} p\left(x_{n+1} \middle| y_{1:n}\right) &= \int f\left(x_{n+1} \middle| x_n\right) p\left(x_n \middle| y_{1:n}\right) dx_n, \\ p\left(x_{n+1} \middle| y_{1:n+1}\right) &= \frac{g\left(y_{n+1} \middle| x_{n+1}\right) p\left(x_{n+1} \middle| y_{1:n}\right)}{p\left(y_{n+1} \middle| y_{1:n}\right)} \end{aligned}$$

- This recursion yields the standard HMM filter and the Kalman filter for linear Gaussian models.

- In our case, this recursion will NOT be used and we will always deal with the joint distributions even if we are only interested in approximating $\left\{p\left(x_n \middle| y_{1:n}\right)\right\}$.

4.4– Recursion for the successive target distributions

- An alternative way to derive the formula is as follows

$$\begin{aligned} p\left(x_{1:n+1} \middle| y_{1:n+1}\right) &\propto p\left(x_{1:n+1}\right) p\left(y_{1:n+1} \middle| x_{1:n+1}\right) \\ &\propto \mu\left(x_1\right) \prod_{k=2}^{n+1} f\left(x_k \middle| x_{k-1}\right) \prod_{k=1}^{n+1} g\left(y_k \middle| x_k\right) \\ &\propto f\left(x_{n+1} \middle| x_n\right) g\left(y_{n+1} \middle| x_{n+1}\right) p\left(x_{1:n}\right) p\left(y_{1:n} \middle| x_{1:n}\right) \\ &\propto f\left(x_{n+1} \middle| x_n\right) g\left(y_{n+1} \middle| x_{n+1}\right) p\left(x_{1:n} \middle| y_{1:n}\right) \end{aligned}$$

4.5– Building Sequentially Monte Carlo Approximations

• Now at time 2, we want to approximate $p\left(x_{1:2}\middle|\, y_{1:2}\right)$. We can also use IS to achieve that by selecting an importance distribution $q_2\left(x_{1:2}\middle|\, y_{1:2}\right)$, using the identity

$$\begin{aligned} p\left(x_{1:2}\middle|\, y_{1:2}\right) &= \frac{w_2\left(x_{1:2}, y_{1:2}\right) q_2\left(x_{1:2}\middle|\, y_{1:2}\right)}{\int w_2\left(x_{1:2}, y_{1:2}\right) q_2\left(x_{1:2}\middle|\, y_{1:2}\right) dx_{1:2}}, \\ w_2\left(x_{1:2}, y_{1:2}\right) &= \frac{p\left(x_{1:2}, y_{1:2}\right)}{q_2\left(x_{1:2}\middle|\, y_{1:2}\right)} \end{aligned}$$

and sampling a large number N of particles

$$X_{1:2}^{(i)} \sim q_2\left(x_{1:2}\middle|\, y_{1:2}\right)$$

to obtain

$$p^N\left(x_{1:2}\middle|\, y_{1:2}\right) = \sum_{i=1}^{N} W_2^{(i)} \delta_{X_{1:2}^{(i)}}\left(x_{1:2}\right) \text{ with } W_2^{(i)} \propto w_2\left(X_{1:2}^{(i)}, y_{1:2}\right).$$

4.5– Building Sequentially Monte Carlo Approximations

• We could repeat this method at each time step n. This would require designing an IS distribution $q_n\left(x_{1:n}\middle|\, y_{1:n}\right)$, sampling N paths $X_{1:n}^{(i)} \sim q_n\left(x_{1:n}\middle|\, y_{1:n}\right)$ and computing the associated weights

$$W_n^{(i)} \propto w_n\left(X_{1:n}^{(i)}, y_{1:n}\right) \text{ where } w_n\left(X_{1:n}^{(i)}, y_{1:n}\right) = \frac{p\left(X_{1:n}^{(i)}, y_{1:n}\right)}{q_n\left(X_{1:n}^{(i)}\middle|\, y_{1:n}\right)}.$$

• In the general case this is NOT a sequential method because the computational complexity increases with the time index n.

• A very simple remark allows us to derive a sequential algorithm. We are going to limit the form of the IS distribution.

4.6– Sequential Importance Sampling

• At time n, we propose not to sample new paths $X_{1:n}^{(i)}$ but to keep the paths $X_{1:n-1}^{(i)}$ which are available at time $n-1$ and just add a component $X_n^{(i)}$. Mathematically, it means that we set

$$\begin{aligned} q_n\left(x_{1:n}\middle|\, y_{1:n}\right) &= \underbrace{q_{n-1}\left(x_{1:n-1}\middle|\, y_{1:n-1}\right)}_{\text{distribution of the paths } X_{1:n-1}^{(i)} \text{ at time } n-1} \\ &\quad \times \underbrace{q_n\left(x_n\middle|\, y_{1:n}, x_{1:n-1}\right)}_{\text{conditional distribution of the new component } X_n^{(i)}} \\ &= q_1\left(x_1\middle|\, y_1\right) \prod_{k=2}^{n} q_k\left(x_k\middle|\, y_{1:k}, x_{1:k-1}\right) \end{aligned}$$

4.6– Sequential Importance Sampling

• In practice, we will actually only used distributions of the form

$$q_n\left(x_n\middle|\, y_{1:n}, x_{1:n-1}\right) = q_n\left(x_n\middle|\, y_n, x_{n-1}\right).$$

This will be justified later but this should be intuitive. Given x_{n-1}, $y_{1:n-1}$ and $x_{1:n-2}$ do not bring any information about X_n.

• We don't have yet a recursive method as IS requires not only to sample the paths $X_{1:n}^{(i)}$ but also requires the computation of the weights

$$W_n^{(i)} \propto w_n\left(X_{1:n}^{(i)}, y_{1:n}\right)$$

4.8– Suboptimal distributions

- A "locally" optimal choice consists of selecting the distribution $q_n\left(x_n\middle|\,y_n, x_{n-1}\right)$ minimizing the variance of

$$\begin{aligned} w_n\left(x_{1:n}, y_{1:n}\right) &= \frac{p\left(y_{1:n}\right) p\left(x_{1:n}\middle|\,y_{1:n}\right)}{q_{n-1}\left(x_{1:n-1}\middle|\,y_{1:n-1}\right) q_n\left(x_n\middle|\,y_n, x_{n-1}\right)} \\ &= \frac{p\left(y_{1:n}\right) p\left(x_{1:n-1}\middle|\,y_{1:n}\right)}{q_{n-1}\left(x_{1:n-1}\middle|\,y_{1:n-1}\right)} \times \frac{p\left(x_n\middle|\,y_n, x_{n-1}\right)}{q_n\left(x_n\middle|\,y_n, x_{n-1}\right)} \end{aligned}$$

conditional upon $x_{1:n-1}$. This is given by

$$q_n\left(x_n\middle|\,y_n, x_{n-1}\right) = p\left(x_n\middle|\,y_n, x_{n-1}\right) = \frac{f\left(x_n\middle|\,x_{n-1}\right) g\left(y_n\middle|\,x_n\right)}{\int f\left(x_n\middle|\,x_{n-1}\right) g\left(y_n\middle|\,x_n\right) dx_n}$$

and

$$w_n\left(x_{1:n}, y_{1:n}\right) \propto w_n\left(x_{1:n-1}, y_{1:n}\right) \times \int f\left(x_n\middle|\,x_{n-1}\right) g\left(y_n\middle|\,x_n\right) dx_n.$$

4.7– Selection of the Importance Distribution

- We know that it is crucial to select a good importance distribution for IS estimate to have reasonable performance.

- The optimal choice is obviously given by

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

but this choice is impossible and we cannot even get a reasonable approximation of it (as in MCMC) because of the sequential design of the importance distribution. For example, remember that $X_1^{(i)} \sim q_1\left(x_1\middle|\,y_1\right)$ whereas at time n, we would love to have $X_1^{(i)} \sim p\left(x_1\middle|\,y_{1:n}\right)$!

4.6– Sequential Importance Sampling

Given $\left\{X_{n-1}^{(i)}, W_{n-1}^{(i)}\right\}$ approximating $p\left(x_{1:n-1}\middle|\,y_{1:n-1}\right)$ at time $n-1$, the algorithm proceeds as follows at time n.

- At time n
  - Sample $X_n^{(i)} \sim q_n\left(x_n\middle|\,y_n, X_{n-1}^{(i)}\right)$ for $i = 1, ..., N$
  - Compute the weights

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

4.6– Sequential Importance Sampling

- The weights satisfy the following recursion

$$\begin{aligned} w_n\left(x_{1:n}, y_{1:n}\right) &= \frac{p\left(x_{1:n}, y_{1:n}\right)}{q_n\left(x_{1:n}\middle|\,y_{1:n}\right)} \\ &= \frac{p\left(x_{1:n-1}, y_{1:n-1}\right)}{q_{n-1}\left(x_{1:n-1}\middle|\,y_{1:n-1}\right)} \times \frac{f\left(x_n\middle|\,x_{n-1}\right) g\left(y_n\middle|\,x_n\right)}{q_n\left(x_n\middle|\,y_n, x_{n-1}\right)} \\ &= w_{n-1}\left(x_{1:n-1}, y_{1:n-1}\right) \times \frac{f\left(x_n\middle|\,x_{n-1}\right) g\left(y_n\middle|\,x_n\right)}{q_n\left(x_n\middle|\,y_n, x_{n-1}\right)} \end{aligned}$$

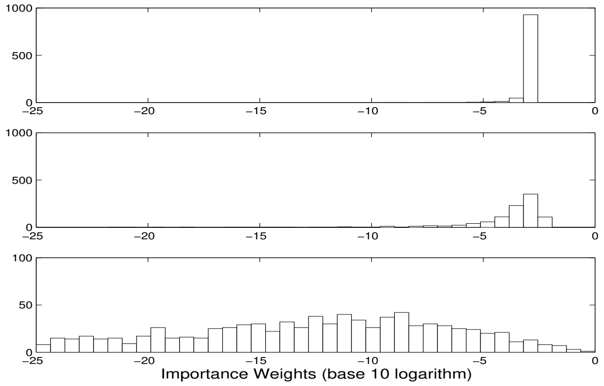
- This implies that

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

- We have designed a SIS scheme of computational complexity $O\left(N\right)$ independent of the time index.

4.10– Simulation Results

- The algorithm performs EXTREMLY poorly! After a few time steps, only a very small number of particles have non negligible weights.



Histograms of the base 10 logarithm of $W_n^{(i)}$ for $n = 1$ (top), $n = 50$ (middle) and $n = 100$ (bottom).

4.9– Application to Stochastic Volatility

- We present a simple application to SV where

$$\begin{aligned} f\left(x_k \mid x_{k-1}\right) &= \mathcal{N}\left(x_{k;}\phi x, \sigma^2\right), \\ g\left(y_k \mid x_k\right) &= \mathcal{N}\left(y_{k;}0, \beta^2 \exp\left(x_k\right)\right). \end{aligned}$$

- We cannot sample from $p\left(x_n \mid y_n, x_{n-1}\right)$ but it is unimodal and we can compute numerically its mode $m_n\left(x_{n-1}\right)$ and use a $t-$distribution with 5 degrees of freedom and scale set as the inverse of the negated second-order of $\log p\left(x_n \mid y_n, x_{n-1}\right)$ evaluated at $m_n\left(x_{n-1}\right)$ and given by

$$\sigma_n^2\left(x_{n-1}\right) = \left(\frac{1}{\sigma^2} + \frac{y_n^2}{2\beta^2} \exp\left(-m_n\left(x_{n-1}\right)\right)\right)^{-1}.$$

4.8– Suboptimal distributions

- A simpler choice consists of selecting

$$q_n\left(x_{1:n} \mid y_{1:n}\right) = p\left(x_{1:n}\right)$$

that is

$$q_n\left(x_1 \mid y_1\right) = \mu\left(x_1\right) \text{ and } q_n\left(x_n \mid y_n, x_{n-1}\right) = f\left(x_n \mid x_{n-1}\right)$$

and

$$w_n\left(x_{1:n}, y_{1:n}\right) = w_{n-1}\left(x_{1:n-1}, y_{1:n-1}\right) \times g\left(y_n \mid x_n\right).$$

- This choice will be extremely poor if the data are very informative and the prior is diffuse.

4.8– Suboptimal distributions

- It is not always possible to use this choice but one can make some approximations.

- For example, one can use an Extended/Unscented Kalman filter to come up with a clever proposal.

- The key is once more that asymptotically (as $N \to \infty$), the Monte Carlo approximation will converge towards the true values.

• We can also use the entropy

$$Ent = -\sum_{i=1}^{N} W_n^{(i)} \log_2 \left(W_n^{(i)} \right)$$

• We have $Ent = \log_2 (N)$ if $W_n^{(i)} = 1/N$ for any i.

• We have $Ent = 0$ if $W_n^{(i)} = 1$ and $W_n^{(j)} = 1$ for $j \neq i$.

• To measure the variation of the weights, we can use the Effective Sample Size (ESS) or the coefficient of variation CV

$$ESS = \left(\sum_{i=1}^{N} \left(W_n^{(i)} \right)^2 \right)^{-1}, \; CV = \left(\frac{1}{N} \sum_{i=1}^{N} \left(N W_n^{(i)} - 1 \right)^2 \right)^{1/2}$$

• We have $ESS = N$ and $CV = 0$ if $W_n^{(i)} = 1/N$ for any i.

• We have $ESS = 1$ and $CV = \sqrt{N-1}$ if $W_n^{(i)} = 1$ and $W_n^{(j)} = 1$ for $j \neq i$.

• *Intuitive KEY idea*: When the variance of the weights $\left\{ W_n^{(i)} \right\}$ is high, we would like to get rid of the particles with low weights (relative to $1/N$) and multiply the particles with high weights.

• The main reason is that 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).

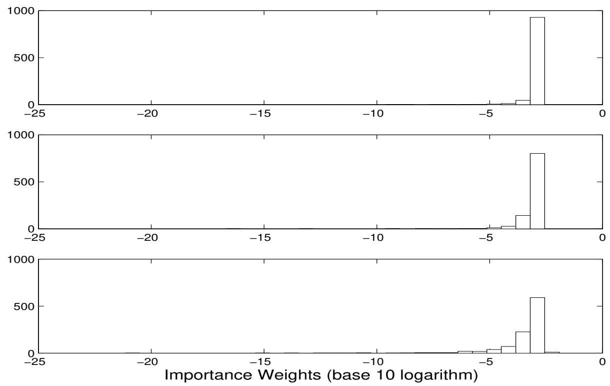
• You want to focus your computational efforts on the “promising” parts of the space.

• You should not be surprised! This algorithm is nothing but an implementation of IS where we severely restrict the structure of the importance distribution.

• As the dimension of the target $p(x_{1:n} | y_{1:n})$ increases over time, the problem is becoming increasingly difficult. In practice, the discrepancy between the target and the IS distribution $q_n (x_{1:n} | y_{1:n})$ can only also increase (on average).

• As n increases the variance of the weights increases (typically geometrically) and the IS approximation collapses.

• You can use any IS distribution you want (even the locally optimal one), the algorithm will collapse.



Histograms of the base 10 logarithm of $W_n^{(i)}$ for $n = 1$ (top), $n = 50$ (middle) and $n = 100$ (bottom).

- At time n
  - Sample $X_n^{(i)} \sim q_n\left(x_n \mid y_n, X_{n-1}^{(i)}\right)$ for $i = 1, ..., N$
  - Compute the weights

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

  - If the variation of the weights is high, resample the particles

$\left\{X_{1:n}^{(i)}, W_n^{(i)}\right\}$ to obtain a new population $\left\{X_{1:n}^{(i)}, 1/N\right\}$.

• This corresponds to perform an approximation of $p^N\left(x_{1:n} \mid y_{1:n}\right)$

$$\sum_{i=1}^{N} \frac{N_n^{(i)}}{N} \delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right) \simeq \sum_{i=1}^{N} W_n^{(i)} \delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right)$$

where $N_n^{(i)}$ is the number of offspring of the particle $X_{1:n}^{(i)}$ and $\sum_{i=1}^{N} N_n^{(i)} = N$.

• The previous scheme is equivalent to sample

$$\left(N_n^{(1)}, ..., N_n^{(N)}\right) \sim \mathcal{M}\left(N; W_n^{(1)}, ..., W_n^{(N)}\right)$$

which is such that $E\left(N_n^{(i)}\right) = N W_n^{(i)}$ but better schemes can be developed.

• If the variation of the weights as measured by ESS, CV or Ent is too high, then we resample the particles.

• The simplest way to resample the particles consists of resampling N times from the current approximation

$$\overline{X}_{1:n}^{(i)} \sim p^N\left(x_{1:n} \mid y_{1:n}\right)$$

where

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

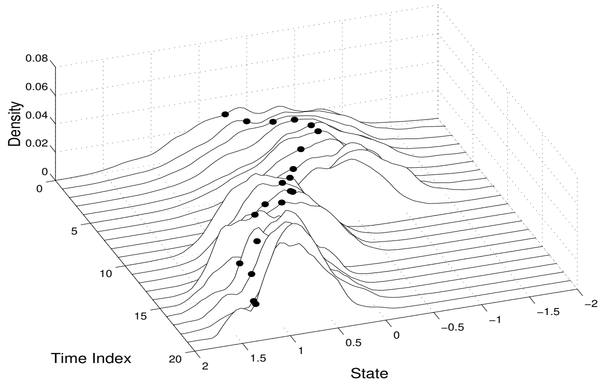
6.1– Summary

- Sequential Importance Sampling is inefficient.

- Resampling is a simple and effective mechanism which mitigates this problem.

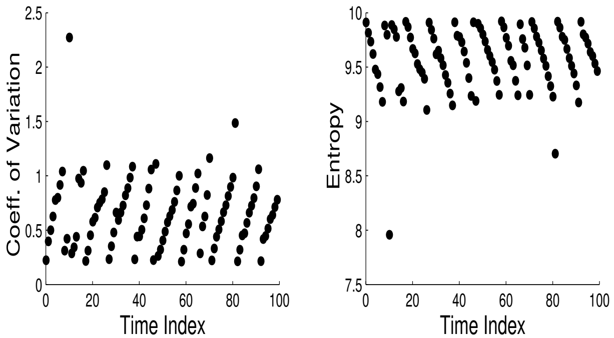
- Next week, we will discuss the design of efficient SMC.

5.1– Resampling



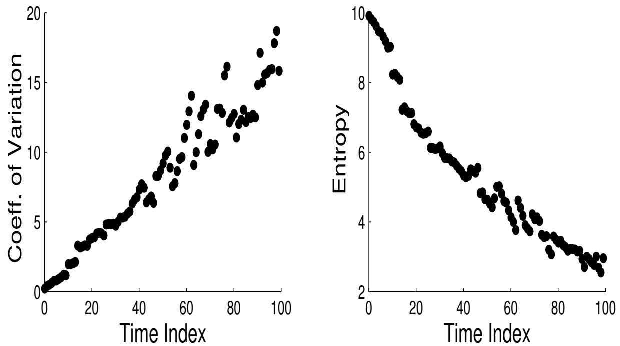
Monte Carlo estimates of the marginal distributions $p\left(x_n \mid y_{1:n}\right)$ and true values of $\{X_n\}$.

5.1– Resampling



Coefficient of Variation and Entropy when Resampling is used.

5.1– Resampling



Coefficient of Variation and Entropy when NO resampling is used.