Stat 535 C - Statistical Computing & Monte Carlo Methods

Lecture 21 - 28th March 2006

Arnaud Doucet

Email: arnaud@cs.ubc.ca

1.1– Outline

- Sequential Importance Sampling.
- Sequential Importance Sampling Resampling for Optimal Filtering.
- Limitations and Generalizations.

2.1– Nonlinear non-Gaussian State-space models

- Nonlinear non-Gaussian state-space model

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

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

- We are interested in the sequence of posterior distributions

$$\begin{aligned} p \left(x_{1:n} | \, y_{1:n} \right) &\propto p \left(x_{1:n} \right) p \left(y_{1:n} | \, x_{1:n} \right) \\ &= \underbrace{\mu \left(x_1 \right) \prod_{k=2}^{n} f \left(x_k | \, x_{k-1} \right)}_{\text{prior}} \underbrace{\prod_{k=1}^{n} g \left(y_k | \, x_k \right)}_{\text{likelihood}}. \end{aligned}$$

2.2– Importance Sampling

- We propose to use IS

$$\begin{aligned} p\left(\left.x_{1:n}\right|y_{1:n}\right) &= \frac{w_n\left(x_{1:n},y_{1:n}\right)q_n\left(\left.x_{1:n}\right|y_{1:n}\right)}{\int w_n\left(x_{1:n},y_{1:n}\right)q_n\left(\left.x_{1:n}\right|y_{1:n}\right)dx_{1:n}}, \\ w_n\left(x_{1:n},y_{1:n}\right) &= \frac{p\left(x_{1:n},y_{1:n}\right)}{q_n\left(\left.x_{1:n}\right|y_{1:n}\right)} \propto \frac{p\left(\left.x_{1:n}\right|y_{1:n}\right)}{q_n\left(\left.x_{1:n}\right|y_{1:n}\right)}. \end{aligned}$$

- If $X_{1:n}^{(i)} \sim q_n\left(\left.x_{1:n}\right|y_{1:n}\right)$ then $\widehat{q}_n^N\left(\left.x_{1:n}\right|y_{1:n}\right) = \frac{1}{N}\sum_{i=1}^N \delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right)$,

$$\begin{aligned} \widehat{p}^N\left(\left.x_{1:n}\right|y_{1:n}\right) &= \sum_{i=1}^N W_n^{(i)}\delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right), \\ \text{where } W_n^{(i)} &\propto w_n\left(X_{1:n}^{(i)},y_{1:n}\right),\ \sum_{i=1}^N W_n^{(i)} = 1. \end{aligned}$$

3.1– 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

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

3.1– Sequential Importance Sampling

- The weights satisfy the following recursion

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

- This implies that

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

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

3.1– 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

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

3.2– Selection of the Importance Distribution

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

- At time n, the optimal choice in terms of minimizing the variance of the weights $\left\{ w_n \left(X_{1:n}^{(i)}, y_{1:n} \right) \right\}$ 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 $X_1^{(i)} \sim p \left(x_1 \middle| y_{1:n} \right)$!

3.3– Suboptimal distributions

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

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

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

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

and

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

3.3– 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.

3.3– Suboptimal distributions

- A simpler choice consists of selecting

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

that is

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

and

$$\begin{aligned}w_n\left(x_{1:n},y_{1:n}\right) &= w_{n-1}\left(x_{1:n-1},y_{1:n-1}\right)\times g\left(\left.y_n\right|x_n\right)\\ &= \prod_{k=1}^{n}g\left(\left.y_k\right|x_k\right).\end{aligned}$$

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

3.4– Application to Stochastic Volatility

- We present a simple application to SV where

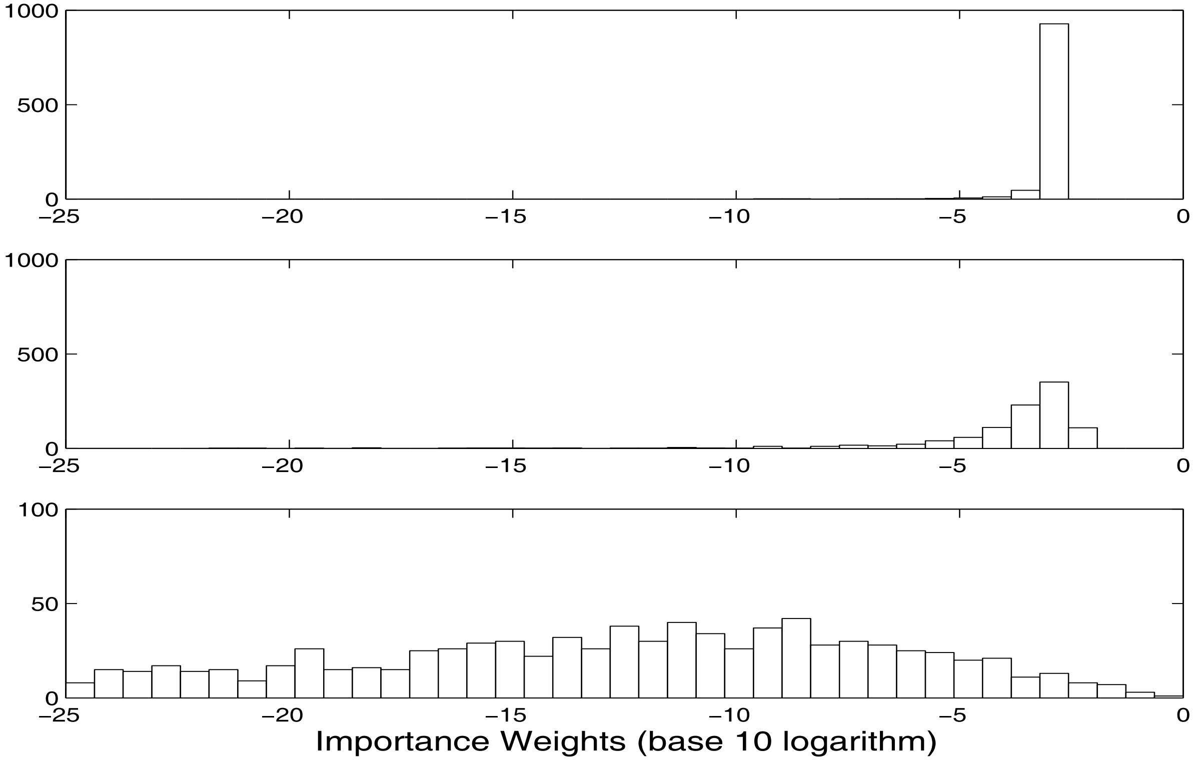
$$\begin{aligned} f\left(\left. x_k \right| x_{k-1} \right) &= \mathcal{N}\left(x_{k;} \phi x, \sigma^2 \right), \\ g\left(\left. y_k \right| x_k \right) &= \mathcal{N}\left(y_{k;} 0, \beta^2 \exp\left(x_k \right) \right). \end{aligned}$$

- We cannot sample from $p\left(\left. x_n \right| y_n, x_{n-1} \right)$ 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 $\log p\left(\left. x_n \right| y_n, x_{n-1} \right)$ evaluated at $m_n\left(x_{n-1} \right)$ and given by

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

3.5– 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).

3.5– Simulation Results

- 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 $p\left(x_{1:n}\middle| y_{1:n}\right)$ increases over time, the problem is becoming increasingly difficult. In practice, the discrepancy between the target and the IS distribution $q_n\left(x_{1:n}\middle| y_{1:n}\right)$ 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.

4.1– Resampling

- *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.

4.1– Resampling

- 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 $j \neq i$.

4.1– Resampling

- 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 $j \neq i$.

4.1– Resampling

- 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

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

where

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

4.1– Resampling

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

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

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

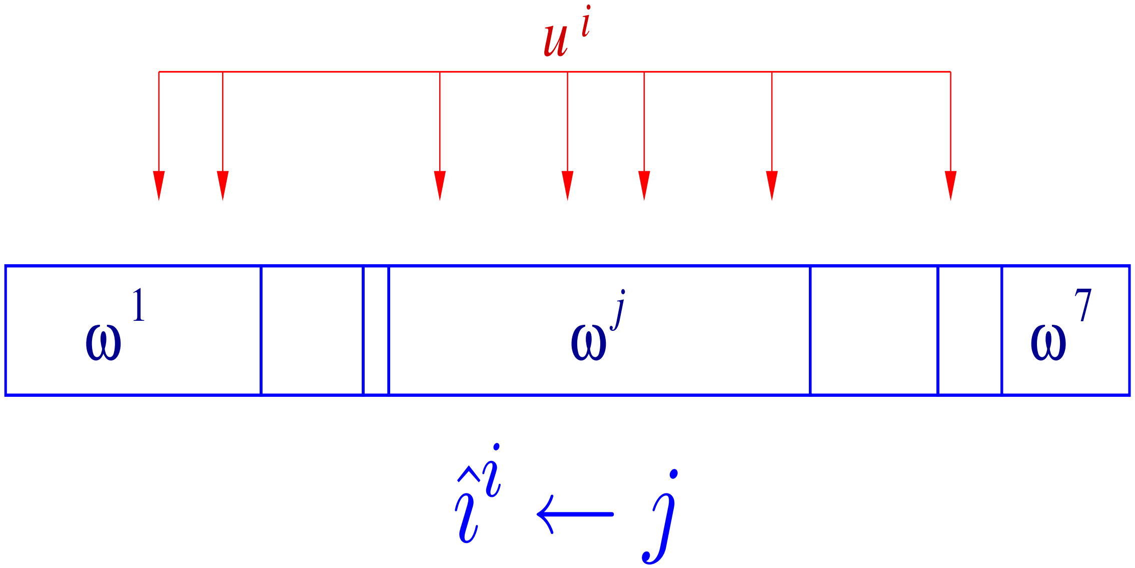
$$\left(N_n^{(1)},...,N_n^{(N)}\right)\sim\mathcal{M}\left(N;W_n^{(1)},...,W_n^{(N)}\right)$$

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

4.2– Multinomial Resampling

- We select N index $\left(\widehat{i}^1, ..., \widehat{i}^N\right)$ amongst $(1, ..., N)$ according to the multinomial of parameters $\left(W_n^{(1)}, ..., W_n^{(N)}\right)$.

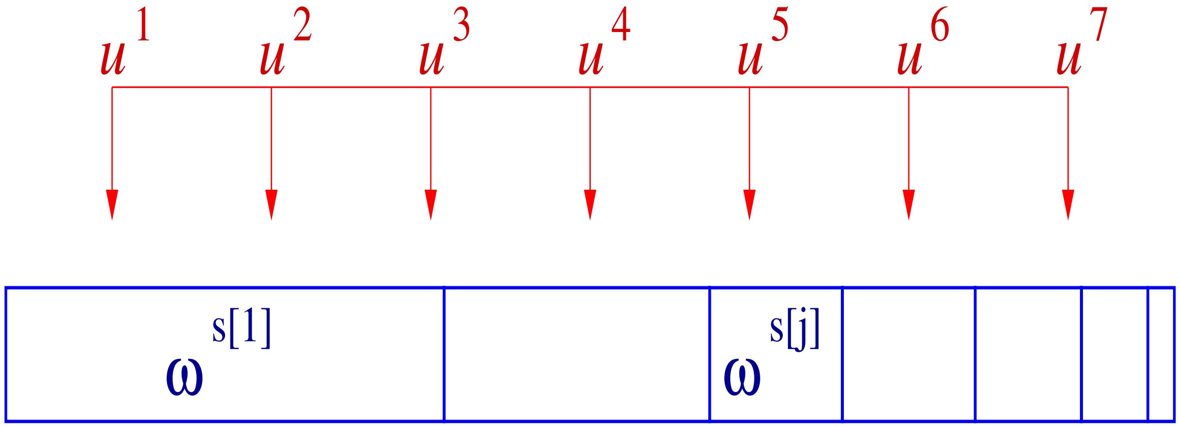
- Practically, we sample $u^i \overset{\text{i.i.d.}}{\sim} U\,[0, 1]$



4.3– Stratified Resampling

- An alternative by Kitagawa consists of selecting

$$u^1 \sim U\left[0, N^{-1}\right],\; u^i = u^1 + \frac{i}{N}$$



$$\hat{i}^i \leftarrow s[j]$$

- There is NO need to sort the weights to apply this procedure.

4.3– Stratified Resampling

- Many alternative algorithms have been proposed for resampling and are typically such that $E\left(N_n^{(i)}\right) = NW_n^{(i)}$ but differ in terms of $var\left(N_n^{(i)}\right)$ and $cov\left(N_n^{(i)}, N_n^{(j)}\right)$.

- We emphasize that there is not much room to improve performance of the resampling step.

4.4– Sequential Importance Sampling Resampling

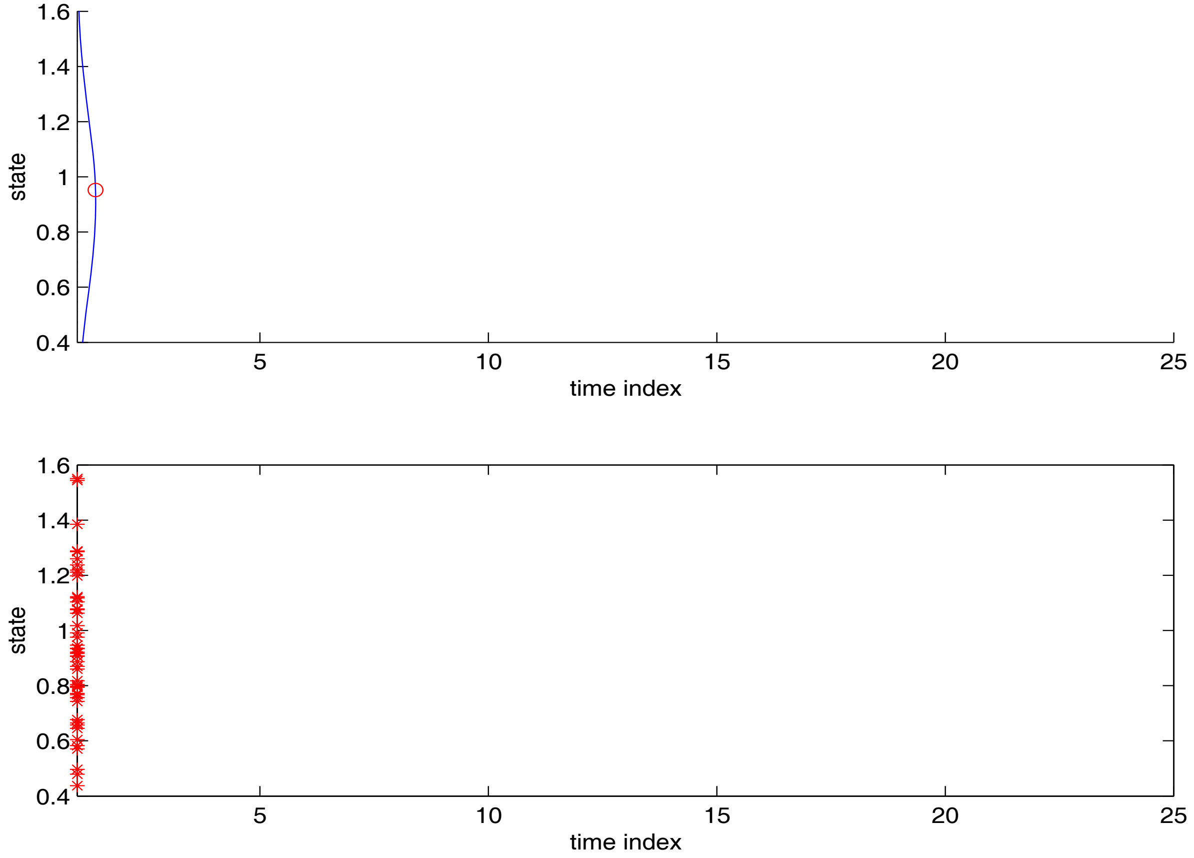
- At time n
  - Sample $X_n^{(i)} \sim q_n \left(x_n \middle| y_n, X_{n-1}^{(i)} \right)$ for $i = 1, ..., N$
  - Compute the weights

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

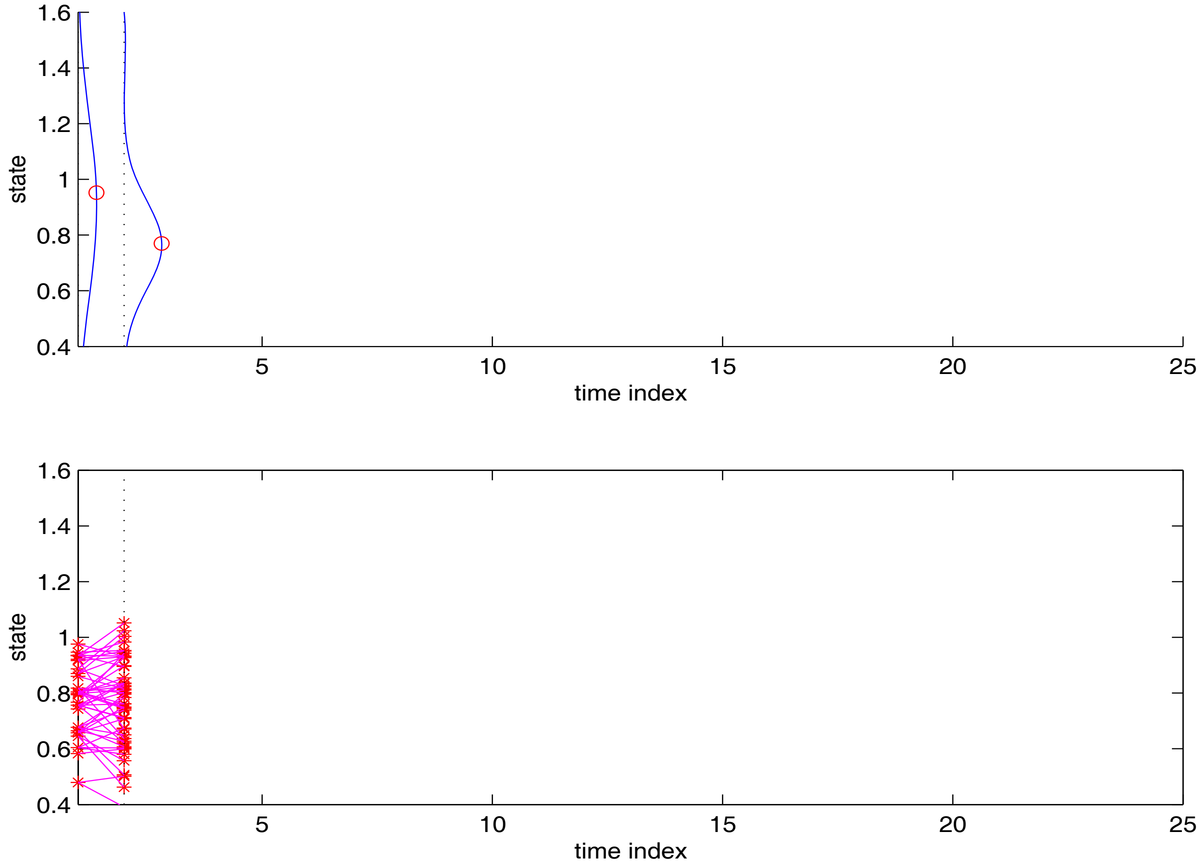
  - If the variation of the weights is high, resample the particles

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

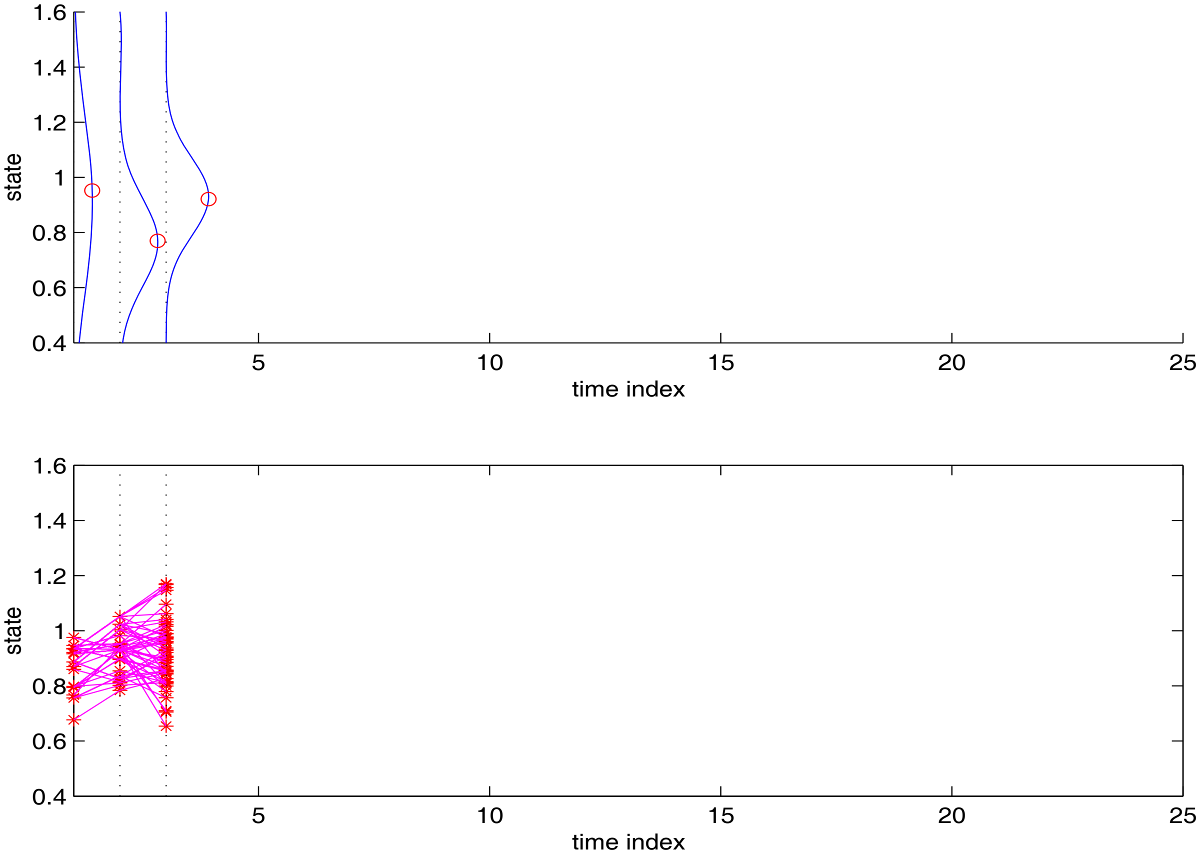
4.5– Experimental Results for Linear Gaussian Model



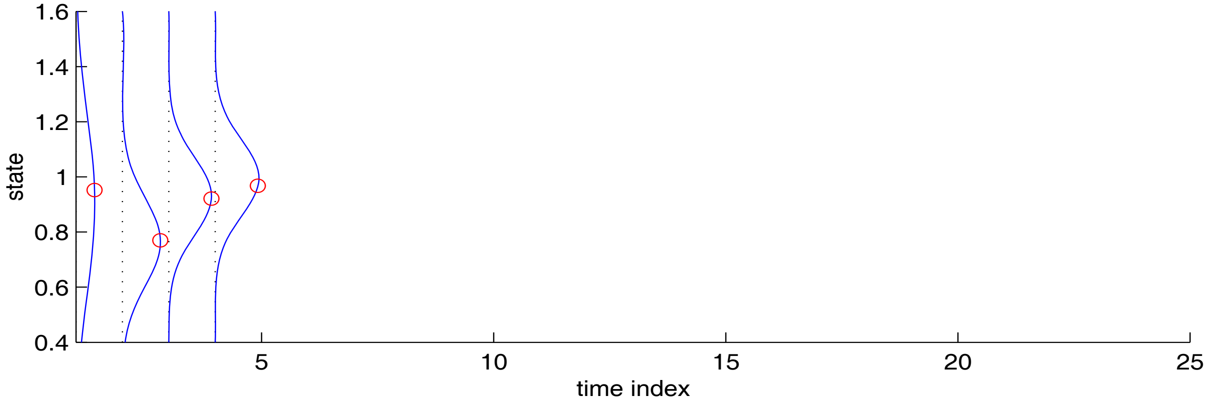
4.5– Experimental Results for Linear Gaussian Model

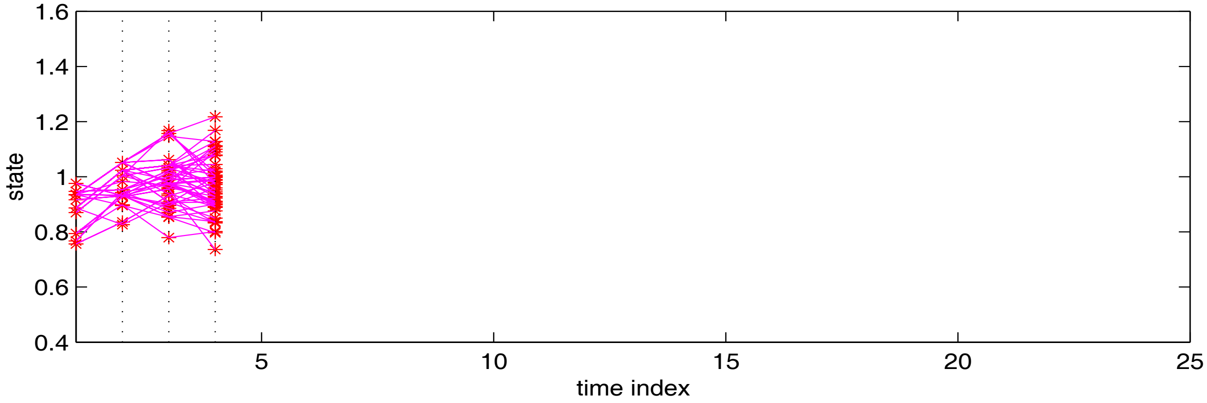


4.5– Experimental Results for Linear Gaussian Model

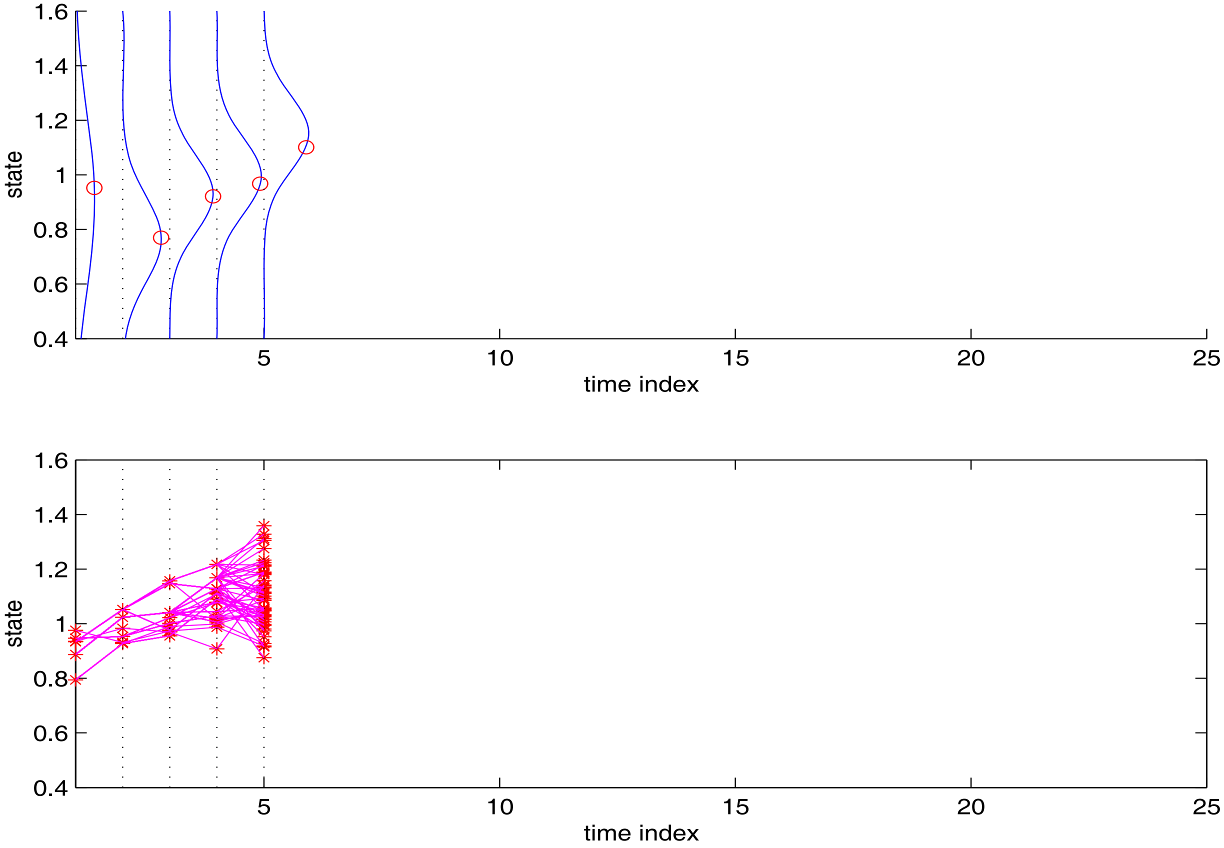


4.5– Experimental Results for Linear Gaussian Model

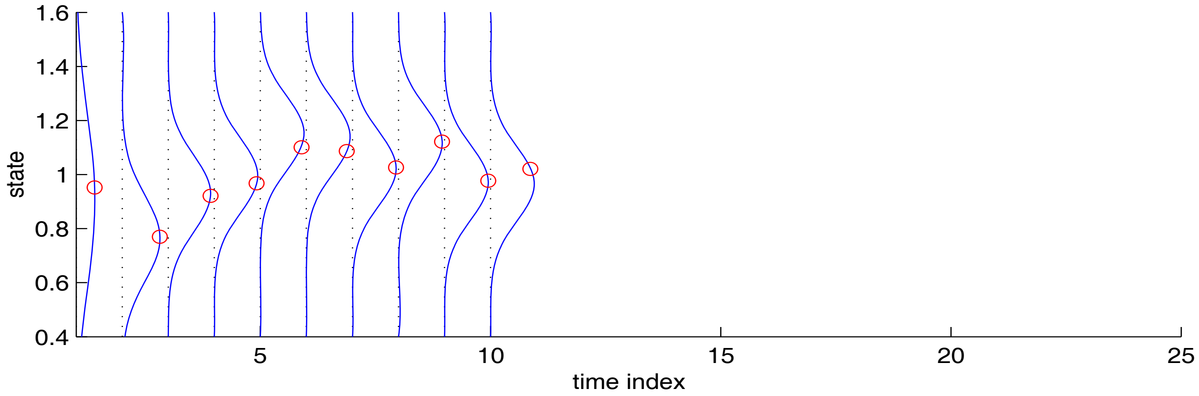


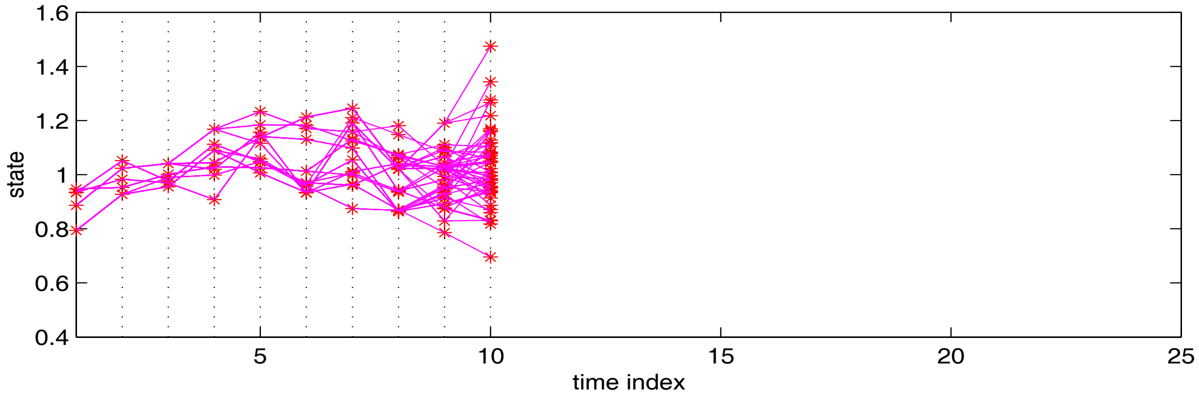


4.5– Experimental Results for Linear Gaussian Model

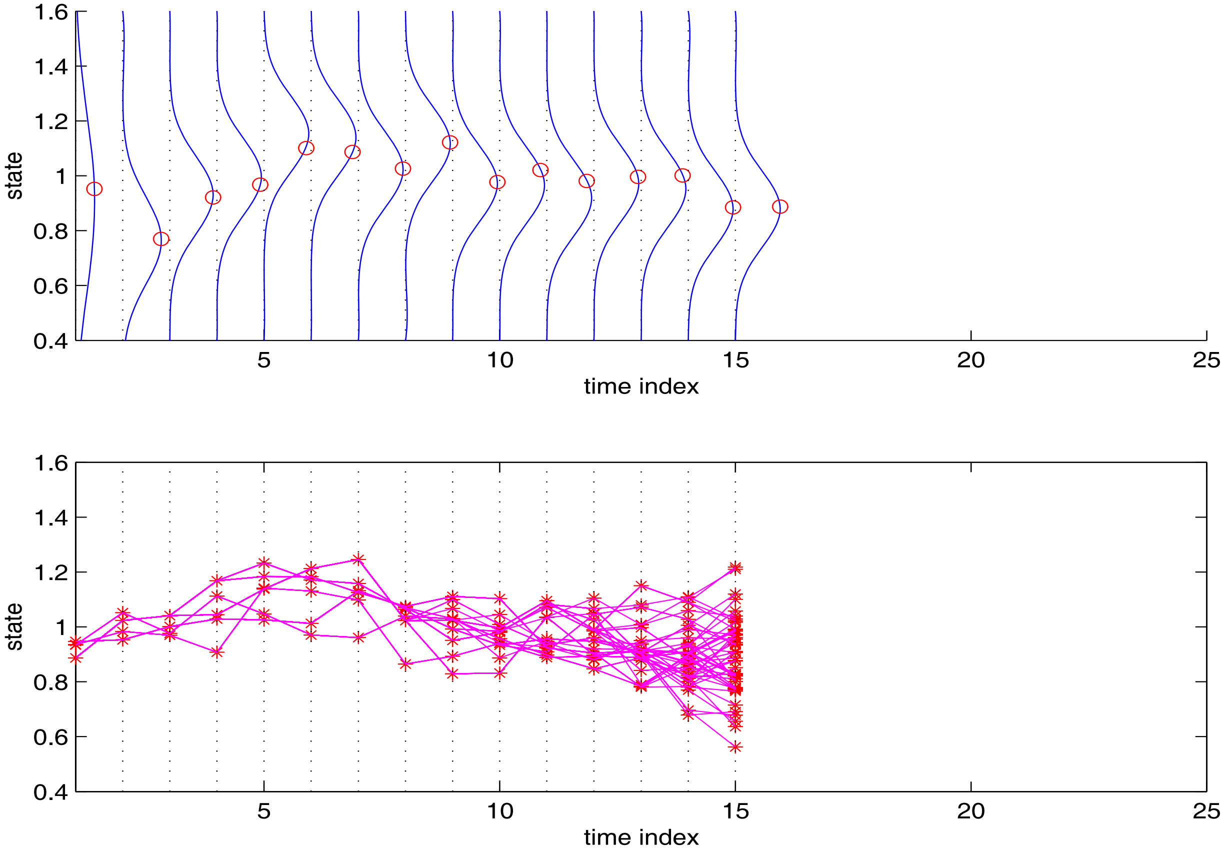


4.5– Experimental Results for Linear Gaussian Model

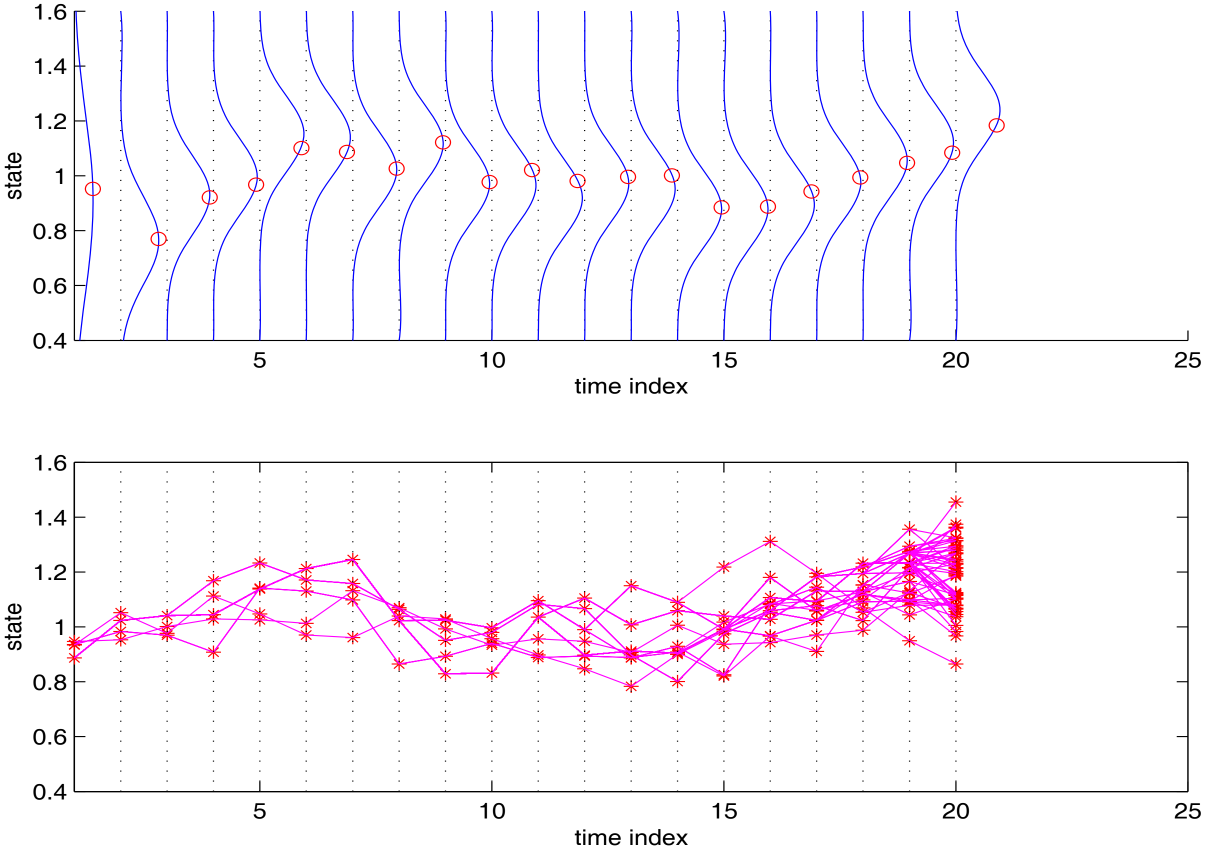




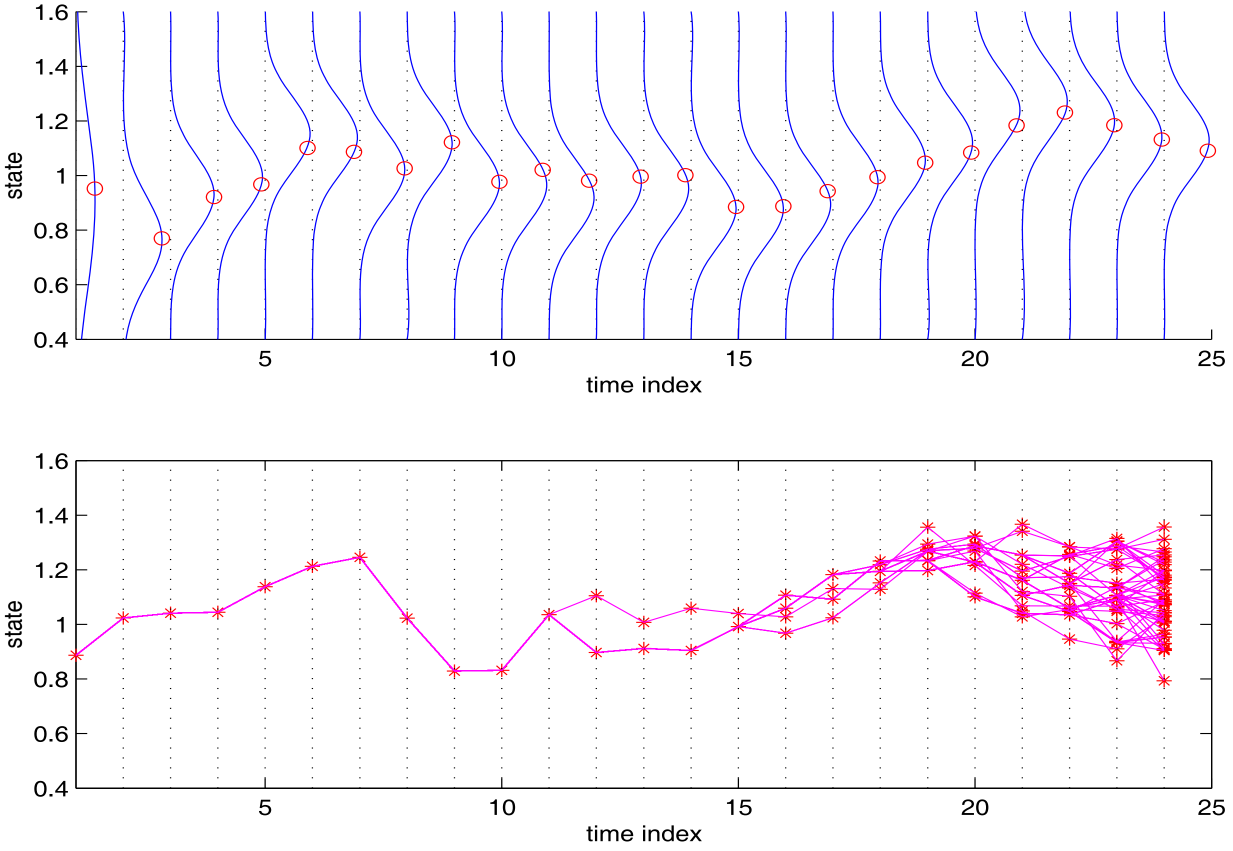
4.5– Experimental Results for Linear Gaussian Model



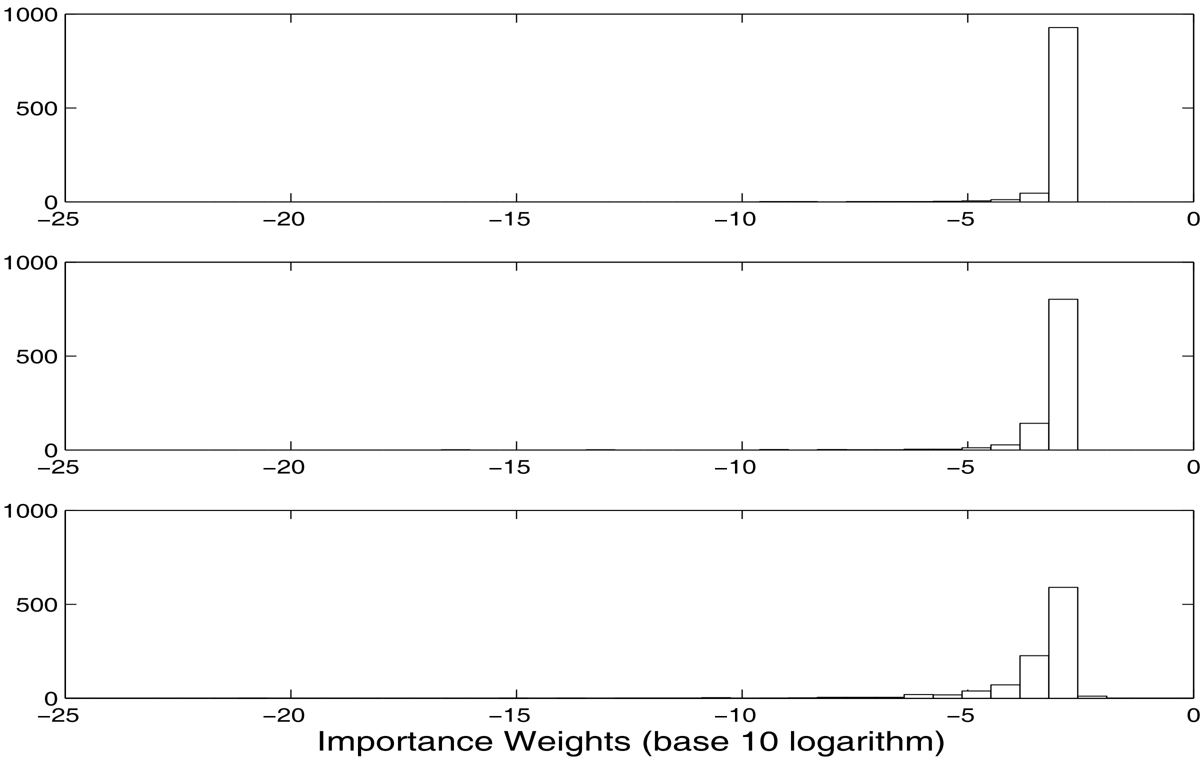
4.5– Experimental Results for Linear Gaussian Model



4.5– Experimental Results for Linear Gaussian Model

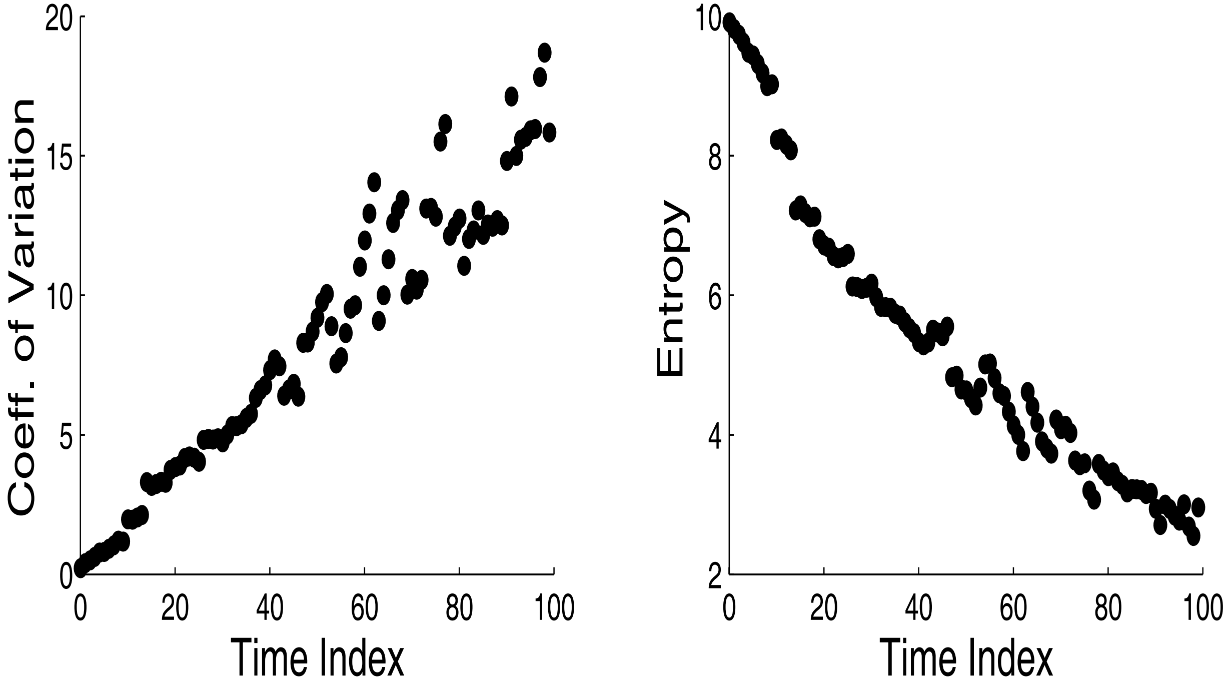


4.6– Experimental Results for Stochastic Volatility Model



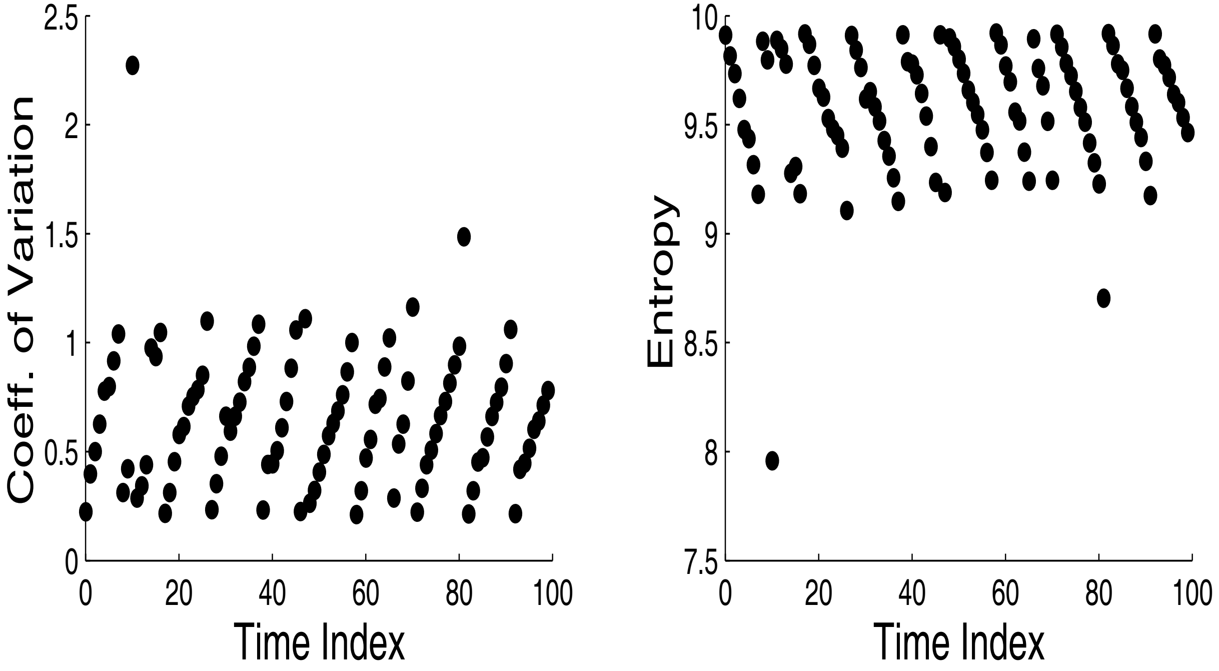
Histograms of the base 10 logarithm of $W_n^{(i)}$ for $n = 1$ (top), $n = 50$ (middle) and $n = 100$ (bottom).

4.6– Experimental Results for Stochastic Volatility Model



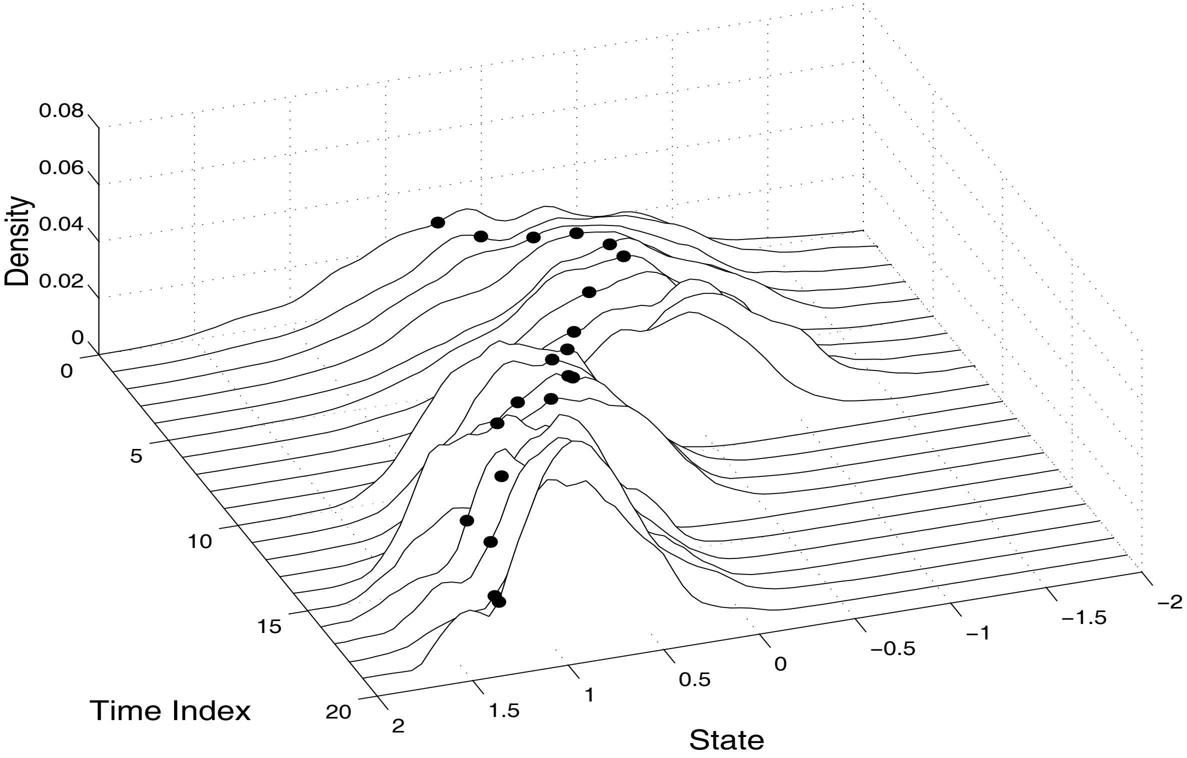
Coefficient of Variation and Entropy when NO resampling is used.

4.6– Experimental Results for Stochastic Volatility Model



Coefficient of Variation and Entropy when Resampling is used.

4.6– Experimental Results for Stochastic Volatility Model



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

5.1– Convergence Results

- Convergence results are beyond the scope of this course but there are many results available.

- In particular we have

$$E\left[\left(\int \varphi\left(x_{1:n}\right)\left(\widehat{p}^{N}\left(\left.dx_{1:n}\right|y_{1:n}\right)-p\left(\left.dx_{1:n}\right|y_{1:n}\right)\right)\right)^{2}\right]\leq\frac{C_{n}}{N}.$$

- It looks like a nice result... but it is rather useless as C_n increases polynomially/exponentially with time.

⇒ To achieve a fixed precision, this would require to use an time-increasing number of particles N.

5.1– Convergence Results

- One cannot hope to estimate with a fixed precision a target distribution of increasing dimension.

- So at best, we can expect results of the following form

$$E\left[\left(\int \varphi\left(x_{n-L+1:n}\right)\left(\widehat{p}^{N}\left(\left.dx_{n-L+1:n}\right|y_{1:n}\right)-p\left(\left.dx_{n-L+1:n}\right|y_{1:n}\right)\right)\right)^{2}\right]\leq\frac{C_L}{N}$$

IF the model has nice forgetting/mixing properties, i.e.

$$\left\|p\left(\left.x_n\right|y_{2:n},x_1\right)-p\left(\left.x_n\right|y_{2:n},x_1'\right)\right\|\leq D\lambda^{n-1}$$

with $\lambda<1$.

- Under such assumptions, there is no accumulation of errors over time.

5.1– Convergence Results

- *Summary*: You can only expect to approximate the "most recent" marginals $p\left(x_{n-L+1:n}\middle|\, y_{1:n}\right)$ but NOT the joint distributions $p\left(x_{1:n}\middle|\, y_{1:n}\right)$.

- This seems rather limited but in most real-world applications we are only interested in the so-called filtering distribution $p\left(x_n\middle|\, y_{1:n}\right)$ and we can also use the property to estimate smoothing distributions

$$p\left(x_k\middle|\, y_{1:n}\right) \simeq p\left(x_k\middle|\, y_{1:k+L}\right)$$

if the system has ergodic properties. Finally we have

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

where

$$p\left(y_k\middle|\, y_{1:k-1}\right) = \int g\left(y_k\middle|\, x_k\right) p\left(x_k\middle|\, y_{1:k-1}\right) dx_k.$$

5.1– Convergence Results

- These results also demonstrate that one cannot expect to obtain good performance if the model has static parameters; i.e. if we have

$$X_1 \sim \mu, \;\; X_k | (X_{k-1} = x_{k-1}) \sim f_\theta (\cdot | x_{k-1}) ,$$

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

where $\theta \sim \pi(\theta)$ and we want to estimate $p(x_{1:n}, \theta | y_{1:n})$.

- Indeed the dynamic model $Z_n = (X_n, \theta)$ is not ergodic as

$$f(x', \theta' | x, \theta) = \delta_\theta (\theta') f_\theta (x' | x) .$$

- This is intuitive! At time 1, we sample N particles $\theta^{(i)}$ and these values are never ever modified later on.

5.1– Convergence Results

- At first glance, this is really bad news. SMC appears unable to deal with static parameters.

- A dirty solution consists of adding noise to a fixed parameter to transform it as a time-varying parameter

$$\theta_n = \theta_{n-1} + \varepsilon_n.$$

- This is not clean and we are going to discuss later on a rigorous approach... which requires a “deeper” understanding of SMC.

6.1– Towards General SMC Methods

- The SMC approach can be extended to any sequence of target distributions

$$\pi_n\left(x_{1:n}\right)=\frac{\gamma_n\left(x_{1:n}\right)}{Z_n}.$$

- In particular, we do not require the target distribution to satisfy

$$\pi_n\left(x_{1:n}\right)\propto\mu\left(x_1\right)\prod_{k=2}^{n}f\left(\left.x_k\right|x_{k-1}\right)\prod_{k=1}^{n}g\left(\left.y_k\right|x_k\right).$$

- The only requirement here is that

$$\pi_n\left(x_{1:n-1}\right)>0\Rightarrow\pi_{n-1}\left(x_{1:n-1}\right)>0.$$

6.2– Conditinionally Linear Gaussian State-Space Models

- As an example consider a switching state-space model

$$\begin{aligned} Z_n &= A(X_n) Z_{n-1} + B(X_n) V_n,\ Z_1 \sim \mathcal{N}(0, \Sigma_0),\ V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I) \\ Y_n &= C(X_n) Z_n + D(X_n) W_n,\ W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I) \end{aligned}$$

where X_n is an unobserved Markov process

$$X_1 \sim \mu, \quad X_n | X_{n-1} = x \sim f(\cdot | x).$$

6.2– Conditinionally Linear Gaussian State-Space Models

- We could estimate using SMC

$$
\begin{aligned}
p\left(x_{1:n}, z_{1:n} \middle| y_{1:n}\right) &\propto p\left(x_{1:n}\right) p\left(z_{1:n} \middle| x_{1:n}\right) p\left(y_{1:n} \middle| x_{1:n}, z_{1:n}\right) \\
&= \mu\left(x_1\right) \prod_{k=2}^{n} f\left(x_k \middle| x_{k-1}\right) p\left(z_1\right) \prod_{k=2}^{n} \mathcal{N} \big(z_n; A\left(x_n\right) z_{n-1}, B\left(x_n\right) B^{\mathsf{T}} (\\
&\quad \times \prod_{k=1}^{n} \mathcal{N} \big(y_n; C\left(x_n\right) z_n, D\left(x_n\right) D^{\mathsf{T}}\left(x_n\right) \big)
\end{aligned}
$$

- This fits in the framework discussed previously: $\{X_n, Z_n\}$ is a Markov process and the observations $\{Y_n\}$ are conditionally independent given $\{X_n, Z_n\}$.

6.3– Variance reduction via Rao-Blackwellisation

- However, conditional upon $\{X_n\}$ the model is linear Gaussian. It follows that we have $$p\left(x_{1:n}, z_{1:n} \middle| y_{1:n}\right) = p\left(x_{1:n} \middle| y_{1:n}\right) \underbrace{p\left(z_{1:n} \middle| y_{1:n}, x_{1:n}\right)}_{\text{Gaussian distribution}}$$

and it is only necessary to estimate through SMC the marginal distribution

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

where the likelihood term is given by the Kalman filter.

- We have $p\left(y_{1:n} \middle| x_{1:n}\right) \neq \prod_{k=1}^{n} p\left(y_k \middle| x_k\right)$ but this does not matter! Additionally we could have also a process $\{X_n\}$ which is non-Markovian. As long as we can compute the target up to a normalizing constant then we will be able to apply SMC.

6.4– Sequential Importance Sampling

- We can use sequential importance sampling

$$X_{1:n}^{(i)} \sim q_n\left(x_{1:n}\right)$$

where

$$\begin{aligned} q_n\left(x_{1:n}\right) &= q_{n-1}\left(x_{1:n-1}\right) q_n\left(\left. x_n \right| y_{1:n}, x_{1:n-1}\right) \\ &= q_1\left(x_1\right) \prod_{k=2}^{n} q_k\left(\left. x_k \right| x_{1:k-1}\right) \end{aligned}$$

- Whether the process is Markov or not does not matter whatsoever.

6.4– Sequential Importance Sampling

- We need to compute the weights associated to each particles $X_{1:n}^{(i)}$.

- We have

$$\begin{aligned} w_n\left(x_{1:n}\right) &= \frac{\gamma_n\left(x_{1:n}\right)}{q_n\left(x_{1:n}\right)} \propto \frac{\pi_n\left(x_{1:n}\right)}{q_n\left(x_{1:n}\right)} \\ &= \frac{\pi_{n-1}\left(x_{1:n-1}\right)}{q_{n-1}\left(x_{1:n-1}\right)} \frac{\pi_n\left(x_{1:n}\right)}{\pi_{n-1}\left(x_{1:n-1}\right) q_n\left(\left. x_n\right| x_{1:n-1}\right)} \\ &\propto w_{n-1}\left(x_{1:n-1}\right) \frac{\pi_n\left(x_{1:n}\right)}{\pi_{n-1}\left(x_{1:n-1}\right) q_n\left(\left. x_n\right| x_{1:n-1}\right)}. \end{aligned}$$

- In many cases, we can compute the incremental weight in a computational time independent of n.

6.5– Selection of the Importance Distribution

- We propose to select the importance distribution minimizing the variance of

the importance
weight conditional upon $x_{1:n-1}$.

- We have

$$\begin{aligned} w_n\left(x_{1:n}\right) & \propto & w_{n-1}\left(x_{1:n-1}\right)\frac{\pi_n\left(x_{1:n}\right)}{\pi_{n-1}\left(x_{1:n-1}\right)q_n\left(\left.x_n\right|x_{1:n-1}\right)} \\ & \propto & w_{n-1}\left(x_{1:n-1}\right)\frac{\pi_n\left(x_{1:n-1}\right)}{\pi_{n-1}\left(x_{1:n-1}\right)}\frac{\pi_n\left(\left.x_n\right|x_{1:n-1}\right)}{q_n\left(\left.x_n\right|x_{1:n-1}\right)}\end{aligned}$$

so the (locally) optimal choice is

$$q_n\left(\left.x_n\right|x_{1:n-1}\right)=\pi_n\left(\left.x_n\right|x_{1:n-1}\right)$$

6.6– Sequential Importance Sampling Resampling

- At time n

  - Sample $X_n^{(i)} \sim q_n \left(x_n \middle| X_{1:n-1}^{(i)} \right)$ for $i = 1, ..., N$

  - Compute the weights

$$W_n^{(i)} \propto W_{n-1}^{(i)} \frac{\pi_n \left(X_{1:n}^{(i)} \right)}{\pi_{n-1} \left(X_{1:n-1}^{(i)} \right) q_n \left(X_n^{(i)} \middle| X_{1:n-1}^{(i)} \right)}$$

  - 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 $\left\{ X_{1:n}^{(i)}, 1/N \right\}$.

7.1– Summary

- Sequential Importance Sampling is inefficient.
- Resampling is a simple and effective mechanism which mitigates this problem.
- On thursday, we will discuss some applications of SMC.