Computer Intensive Methods in Mathematical Statistics

Johan Westerborn

Department of mathematics
KTH Royal Institute of Technology
johawes@kth.se

Lecture 7
Sequential Monte Carlo methods III
7 April 2017

Plan of today's lecture

Outline

Last time: sequential MC problems

- In the framework of SMC the principal goal is to generate sequences of weighted samples $(X^i_{0:n}, \omega^i_n)_{i=1}^N$ that can be used for estimating expectations

$$\tau_n = \mathbb{E}_{f_n}(\phi(X_{0:n})) = \int_{\mathsf{X}_n} \phi(x_{0:n}) f_n(x_{0:n})\, dx_{0:n}$$

over spaces X_n of increasing dimension.

- The densities $(f_n)_{n \geq 0}$ are supposed to be known up to normalizing constants only; i.e. for every $n \geq 0$,

$$f_n(x_{0:n}) = \frac{z_n(x_{0:n})}{c_n},$$

where c_n is an unknown constant and z_n is a known positive function on X_n.

Last time: sequential importance sampling (SIS)

- In order to derive such a procedure we proceeded recursively. Assume that we have generated particles $(X^i_{0:n})_{i=1}^N$ from $g_n(x_{0:n})$ such that

$$\sum_{i=1}^N \frac{\omega_n^i}{\sum_{\ell=1}^N \omega_n^\ell} \phi(X^i_{0:n}) \approx \mathbb{E}_{f_n}(\phi(X_{0:n})),$$

  where, as usual, $\omega_n^i = \omega_n(X^i_{0:n}) = z_n(X^i_{0:n})/g_n(X^i_{0:n})$.
- Key trick: use a sequence $(g_k(x_{k+1} \mid x_{0:k}))_{k\geq 0}$ of conditional distributions to specify the instrumental distribution:

$$g_n(x_{0:n}) = g_0(x_0) \prod_{k=0}^{n-1} g_k(x_{k+1} \mid x_{0:k}).$$

Last time: SIS (cont.)

- Consequently, $X^i_{0:n+1}$ and ω^i_{n+1} can be generated by
  1. keeping the previous $X^i_{0:n}$,
  2. simulating $X^i_{n+1} \sim g_n(x_{n+1} \mid X^i_{0:n})$,
  3. setting $X^i_{0:n+1} = (X^i_{n+1}, X^i_{0:n})$, and
  4. computing

$$
\begin{aligned}
\omega^i_{n+1} &= \frac{z_{n+1}(X^i_{0:n+1})}{g_{n+1}(X^i_{0:n+1})} \\
&= \frac{z_{n+1}(X^i_{0:n+1})}{z_n(X^i_{0:n}) g_n(X^i_{n+1} \mid X^i_{0:n})} \times \frac{z_n(X^i_{0:n})}{g_n(X^i_{0:n})} \\
&= \frac{z_{n+1}(X^i_{0:n+1})}{z_n(X^i_{0:n}) g_n(X^i_{n+1} \mid X^i_{0:n})} \times \omega^i_n .
\end{aligned}
$$

Last time: SIS (cont.)

- So, by running the SIS algorithm we have updated

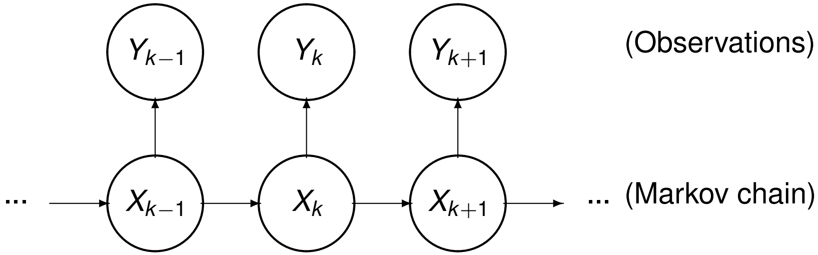
$$(\omega_n^i, X_{0:n}^i)_{i=1}^N \rightsquigarrow (\omega_{n+1}^i, X_{0:n+1}^i)_{i=1}^N$$

  by only adding a component X_{n+1}^i to $X_{0:n}^i$ and updating recursively the weights.
- At time zero the algorithm is initialized by standard importance sampling.
- We note that for each n, an unbiased estimate of c_n can, as usual, be obtained through the average weight:

$$\frac{1}{N}\sum_{i=1}^N \omega_n^i \approx c_n.$$

Filtering in HMMs

- Graphically:



$$
\begin{aligned}
Y_k \mid X_k = x_k &\sim p(y_k \mid x_k) && \text{(observation density)} \\
X_{k+1} \mid X_k = x_k &\sim q(x_{k+1} \mid x_k) && \text{(transition density)} \\
X_0 &\sim \chi(x_0) && \text{(initial distribution)}
\end{aligned}
$$

Filtering in HMMs

- Given a sequence of observed values $(y_n)_{n\geq 0}$, we want to estimate sequentially the filter means

$$\tau_n = \mathbb{E}(X_n | Y_{0:n} = y_{0:n})$$
$$= \int \underbrace{x_n}_{\phi(x_{0:n})} \underbrace{\frac{\chi(x_0)p(y_0|x_0)\prod_{k=0}^{n-1} q(x_{k+1}|x_k)p(y_{k+1}|x_{k+1})}{L_n(y_{0:n})}}_{z_n(x_{0:n})/c_n} \, dx_{0:n}$$

in a single sweep of the data as new observations become available.

Filtering in HMMs

- To obtain a SIS implementation, we simply set

$$g_{n+1}(x_{n+1} \mid x_{0:n}) = q(x_{n+1} \mid x_n),$$

- This implies

$$\begin{aligned}\omega_{n+1}^i &= \frac{z_{n+1}(X_{0:n+1}^i)}{z_n(X_{0:n}^i) g_{n+1}(X_{n+1}^i \mid X_{0:n}^i)} \omega_n^i \\ &= \frac{\chi(X_0^i) p(y_0 | X_0^i) \prod_{k=0}^{n} q(X_{k+1}^i | X_k^i) p(y_{k+1} | X_{k+1}^i)}{\chi(X_0^i) p(y_0 | X_0^i) \{\prod_{k=0}^{n-1} q(X_{k+1}^i | X_k^i) p(y_{k+1} | X_{k+1}^i)\} q(X_{n+1}^i | X_n^i)} \omega_n^i \\ &= p(y_{n+1} \mid X_{n+1}^i) \omega_n^i.\end{aligned}$$

Linear/Gaussian HMM, SIS implementation (cont'd)

- Consider a linear Gaussian HMM:

$$\begin{aligned} Y_k &= X_k + S\varepsilon_k, & &\sim p(y_k|x_k) \\ X_{k+1} &= AX_k + R\epsilon_{k+1}, & &\sim q(x_{k+1}|x_k) \\ X_0 &= R/(1-A^2)\epsilon_0, & &\sim \chi(x_0) \end{aligned}$$

  where $|A| < 1$ and (ϵ_k) and (ε_k) are Gaussian noise.
- then each update $(X_n^i, \omega_n^i) \rightsquigarrow (X_{n+1}^i, \omega_{n+1}^i)$ involves
  1. drawing $X_{n+1}^i \sim \mathrm{N}(AX_n^i, R^2)$,
  2. setting $\omega_{n+1}^i = \mathrm{N}(Y_{n+1}; X_{n+1}^i, S^2)\omega_n^i$.

Linear/Gaussian HMM, SIS implementation (cont'd)

In Matlab:

```
N = 1000;
n = 60;
tau = zeros(1,n); % vector of estimates
p = @(x,y) normpdf(y,x,S); % observation density, for weights
part = R*sqrt(1/(1 - A^2))*randn(N,1); % initialization
w = p(part,Y(1));
tau(1) = sum(part.*w)/sum(w);
for k = 1:n, % main loop
    part = A*part + R*randn(N,1); % mutation
    w = w.*p(part,Y(k + 1)); % weighting
    tau(k + 1) = sum(part.*w)/sum(w); % estimation
end
```

Linear/Gaussian HMM, SIS implementation (cont.)

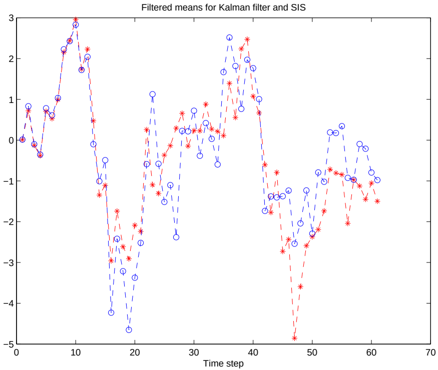


Figure: Comparison of SIS (○) with exact values (∗) provided by the Kalman filter (possible only for linear/Gaussian models).

Linear/Gaussian HMM, SIS implementation (cont.)

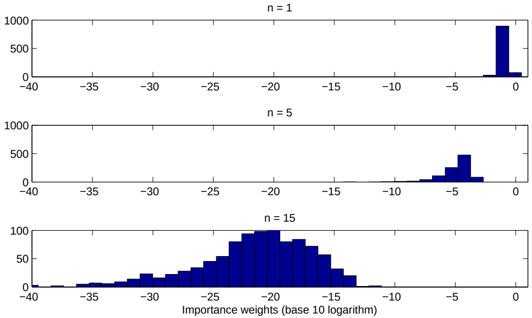


Figure: Distribution of importance weights ☹

Outline

Interlude: IS with resampling (ISR)

- Having at hand a weighted sample $(X^i, \omega(X^i))_{i=1}^N$ approximating f, a uniformly weighted sample can be formed by resampling, with replacement, new variables $(\tilde{X}^i)_{i=1}^N$ from $(X^i)_{i=1}^N$ according to the weights $(\omega(X^i))_{i=1}^N$.
- This can be achieved using the following algorithm:

Data: $(X^i, \omega(X^i))_{i=1}^N$
Result: $(\tilde{X}^i)_{i=1}^N$
for $i = 1 \to N$ **do**
  set $I^i \leftarrow j$ w. pr. $\dfrac{\omega(X^j)}{\sum_{\ell=1}^N \omega(X^\ell)}$;
  set $\tilde{X}^i \leftarrow X^{I^i}$;
end

Interlude: ISR (cont.)

- The ISR algorithm replaces

$$\sum_{i=1}^{N} \frac{\omega(X^i)}{\sum_{\ell=1}^{N} \omega(X^\ell)} \phi(X^i) \quad \text{by} \quad \frac{1}{N} \sum_{i=1}^{N} \phi(\tilde{X}^i).$$

- The resampling step does not add bias to the estimator in the following sense.

Theorem (unbiasedness, multinomial resampling)

For all $N \geq 1$ it holds that

$$\mathbb{E}\left(\frac{1}{N} \sum_{i=1}^{N} \phi(\tilde{X}^i) \right) = \mathbb{E}\left(\sum_{i=1}^{N} \frac{\omega(X^i)}{\sum_{\ell=1}^{N} \omega(X^\ell)} \phi(X^i) \right).$$

Outline

Outline

SIS + ISR

- A simple—but revolutionary!—idea: duplicate/kill particles with large/small weights! (Gordon *et al.*, 1993)
- The most natural approach to such selection is to simply draw, as in the ISR algorithm, new particles $(\tilde{X}^i_{0:n})_{i=1}^N$ among the SIS-produced particles $(X^i_{0:n})_{i=1}^N$ with probabilities given by the normalized importance weights.
- Formally, this amounts to set, for $i = 1, 2, \ldots, N$,

$$\tilde{X}^i_{0:n} = X^j_{0:n} \text{ w. pr. } \frac{\omega^j_n}{\sum_{\ell=1}^N \omega^\ell_n}.$$

SIS + ISR (cont.)

- After this, the resampled particles $(\tilde{X}^i_{0:n})_{i=1}^N$ are assigned equal weights $\tilde{\omega}^i_n = 1$ and we replace

$$\sum_{i=1}^N \frac{\omega^i_n}{\sum_{\ell=1}^N \omega^\ell_n} \phi(X^i_{0:n}) \quad \text{by} \quad \frac{1}{N} \sum_{i=1}^N \phi(\tilde{X}^i_{0:n}).$$

- As established above, resampling does not add bias:

Corollary

For all $N \geq 1$ and $n \geq 0$,

$$\mathbb{E}\left(\frac{1}{N}\sum_{i=1}^N \phi(\tilde{X}^i_{0:n})\right) = \mathbb{E}\left(\sum_{i=1}^N \frac{\omega^i_n}{\sum_{\ell=1}^N \omega^\ell_n} \phi(X^i_{0:n})\right).$$

... = SIS with resampling (SISR)

- After selection, we proceed with standard SIS and move the selected particles $(\tilde{X}^i_{0:n})^N_{i=1}$ according to $g_n(x_{n+1} \mid x_{0:n})$.
- The full scheme goes as follows. Given $(X^i_{0:n}, \omega^i_n)^N_{i=1}$,
  1. (selection) draw, with replacement, $(\tilde{X}^i_{0:n})^N_{i=1}$ among $(X^i_{0:n})^N_{i=1}$ according to probabilities $(\omega^i_n / \sum_{\ell=1}^N \omega^\ell_n)^N_{i=1}$
  2. (mutation) draw, for all i, $X^i_{n+1} \sim g_n(x_{n+1} \mid \tilde{X}^i_{0:n})$,
  3. set, for all i, $X^i_{0:n+1} = (\tilde{X}^i_{0:n}, X^i_{n+1})$, and
  4. set, for all i,

$$\omega^i_{n+1} = \frac{z_{n+1}(X^i_{0:n+1})}{z_n(X^i_{0:n}) g_n(X^i_{n+1} \mid X^i_{0:n})}.$$

SISR (cont.)

- At every time step n, both

$$\sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} \phi(X_{0:n}^i) \quad \text{and} \quad \frac{1}{N} \sum_{i=1}^{N} \phi(\tilde{X}_{0:n}^i)$$

are valid estimators of τ_n.
- The former incorporates however somewhat less randomness and has thus slightly lower variance.

SISR, estimation of c_n

- When the particles are resampled, the weight average $\frac{1}{N}\sum_{i=1}^{N}\omega_n^i$ is no longer a valid MC estimator of the normalizing constant c_n.
- Instead, one takes

$$c_{N,n}^{\text{SISR}} = \prod_{k=0}^{n}\left(\frac{1}{N}\sum_{i=1}^{N}\omega_k^i\right).$$

- Remarkably, the estimator $c_{N,n}^{\text{SISR}}$ is unbiased!

Theorem

For all $n \geq 0$ and $N \geq 1$,

$$\mathbb{E}\left(c_{N,n}^{SISR}\right) = c_n.$$

Linear/Gaussian HMM, SISR implementation

```
N = 1000;
n = 60;
tau = zeros(1,n); % vector of filter means
w = zeros(N,1);
p = @(x,y) normpdf(y,x,S); % observation density, for weights
part = R*sqrt(1/(1 - A^2))*randn(N,1); % initialization
w = p(part,Y(1));
tau(1) = sum(part.*w)/sum(w);
for k = 1:n, % main loop
    part = A*part + R*randn(N,1); % mutation
    w = p(part,Y(k + 1)); % weighting
    tau(k + 1) = sum(part.*w)/sum(w); % estimation
    ind = randsample(N,N,true,w); % selection
    part = part(ind);
end
```

Linear/Gaussian HMM, SISR implementation (cont'd)

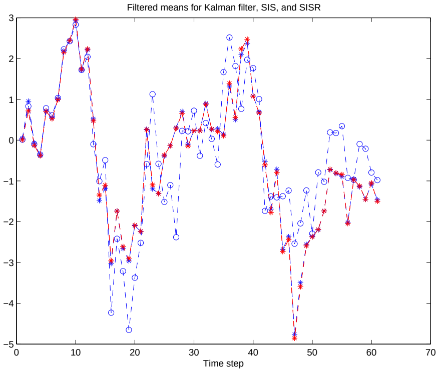
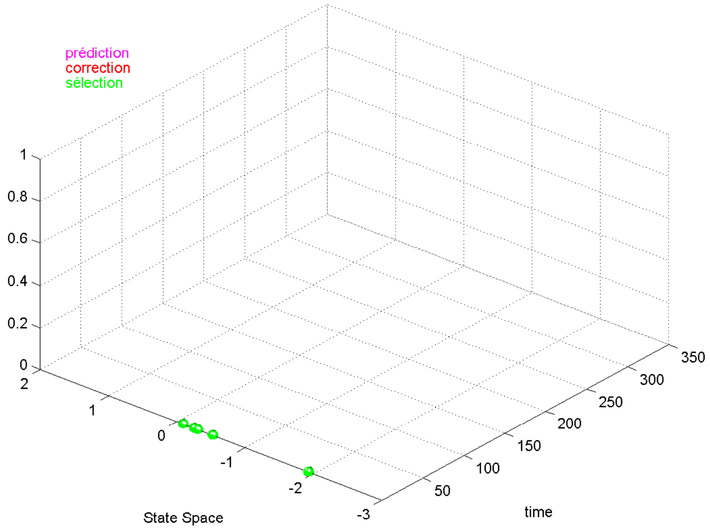


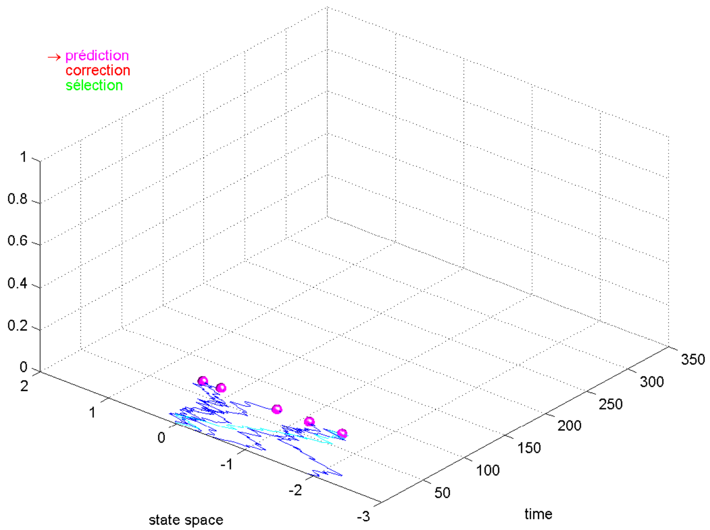
Figure: Comparison of SIS (∘) and SISR (∗, blue) with exact values (∗, red) provided by the Kalman filter.

Film time! ☺

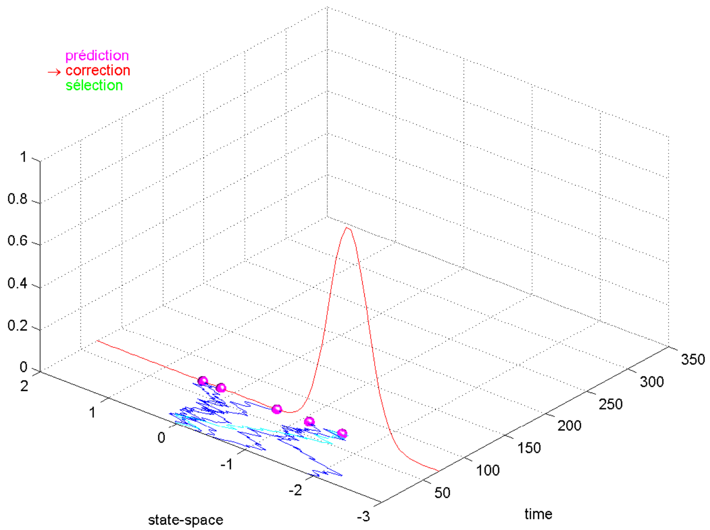
SIS with multinomial selection



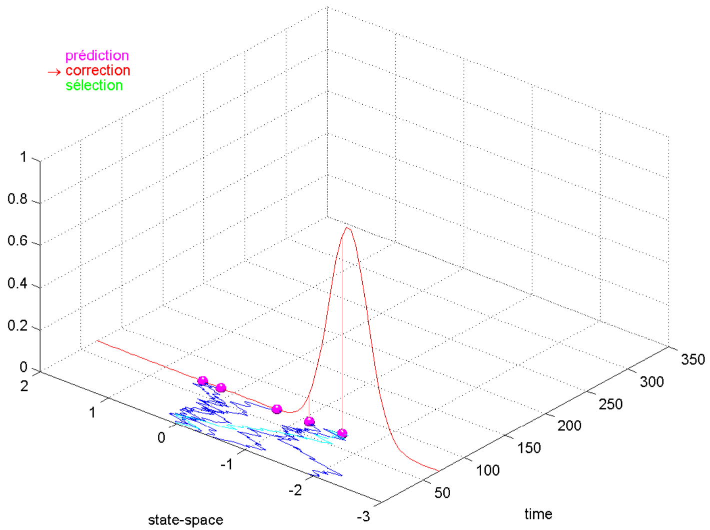
SIS with multinomial selection



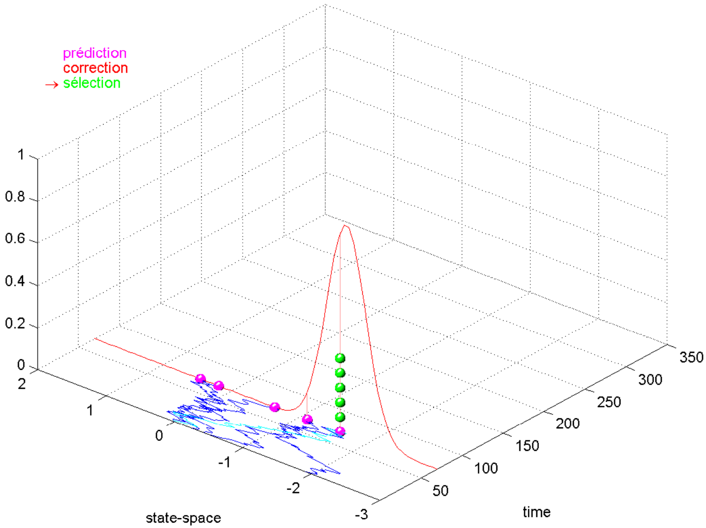
SIS with multinomial selection



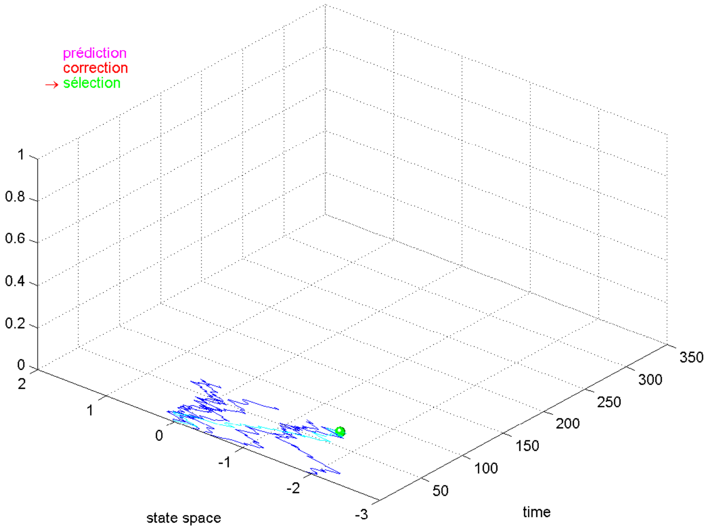
SIS with multinomial selection



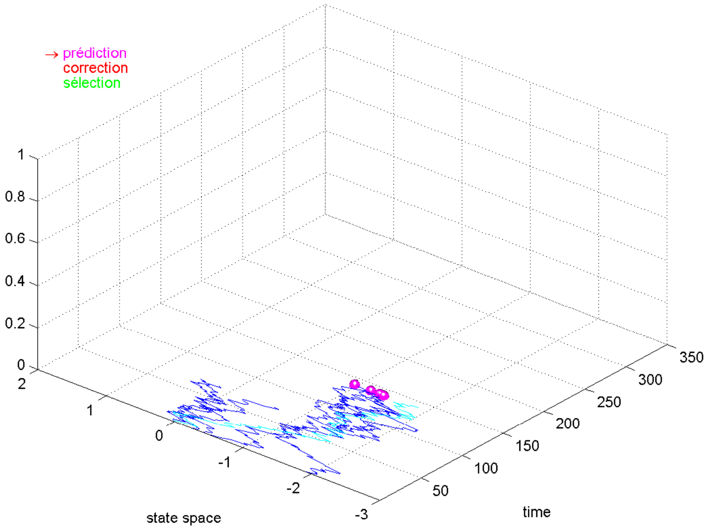
SIS with multinomial selection



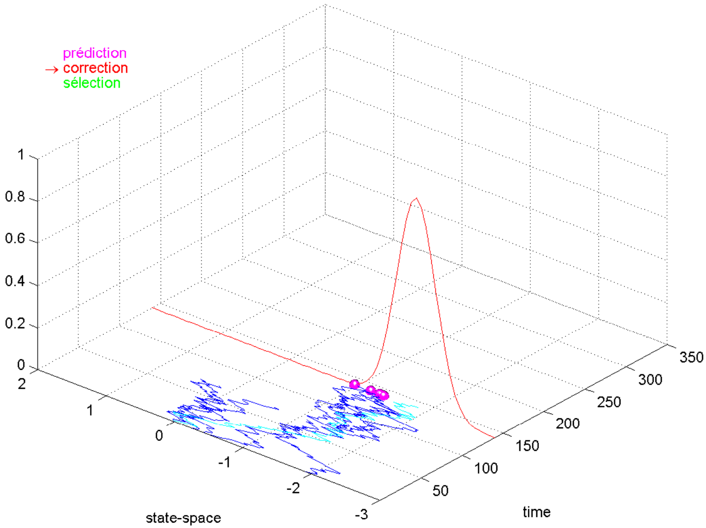
SIS with multinomial selection



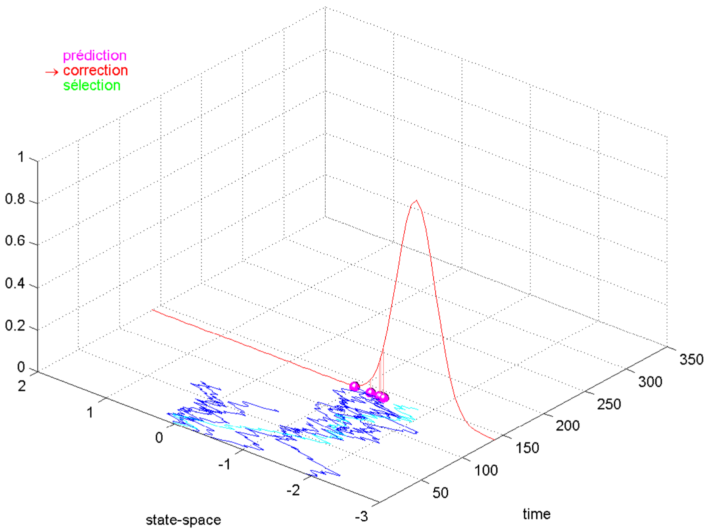
SIS with multinomial selection



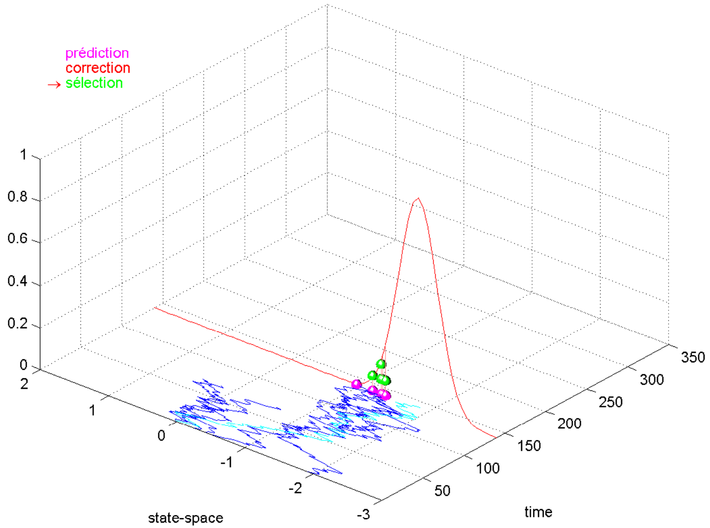
SIS with multinomial selection



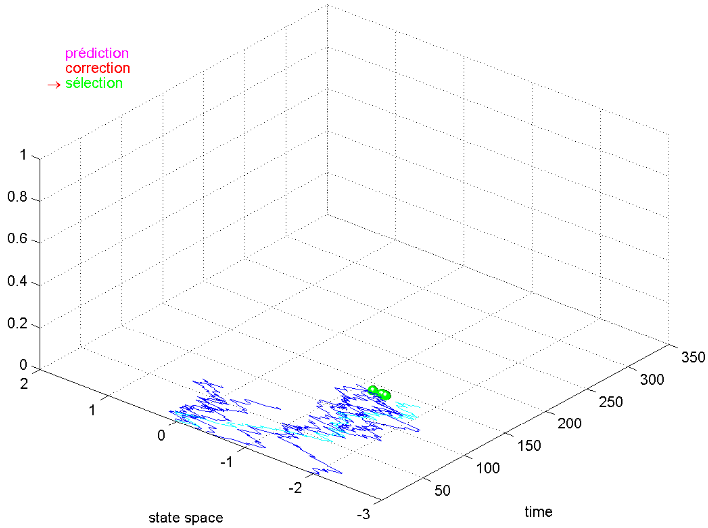
SIS with multinomial selection



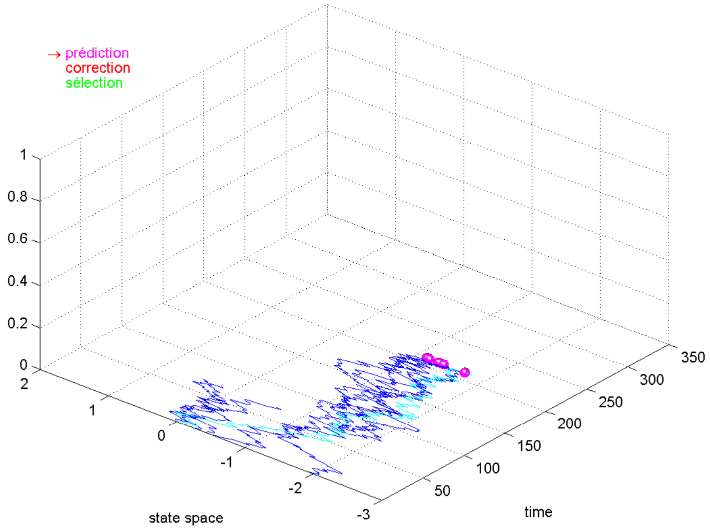
SIS with multinomial selection



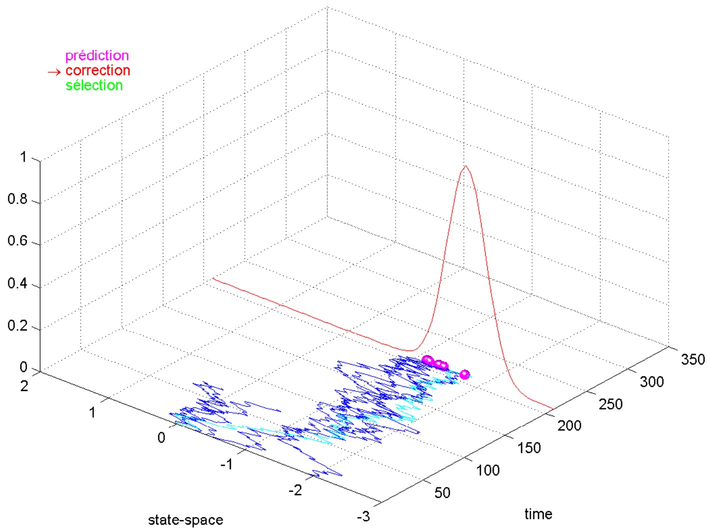
SIS with multinomial selection



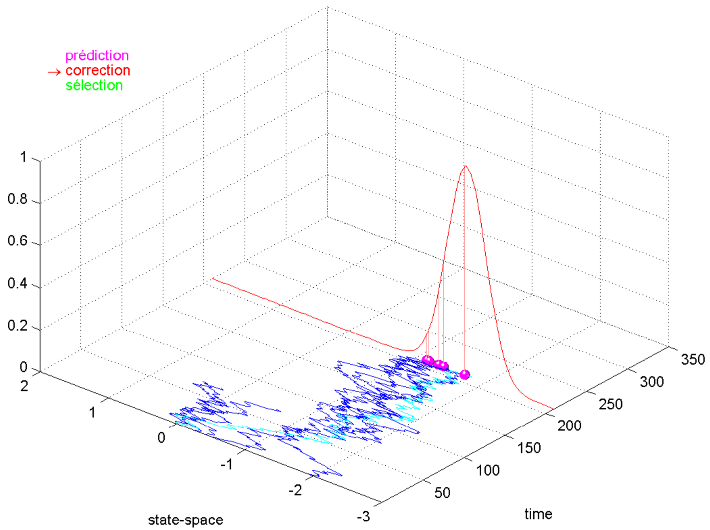
SIS with multinomial selection



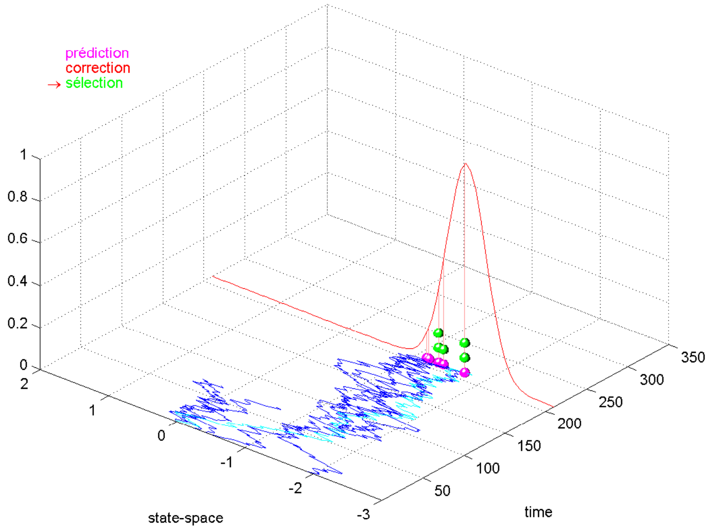
SIS with multinomial selection



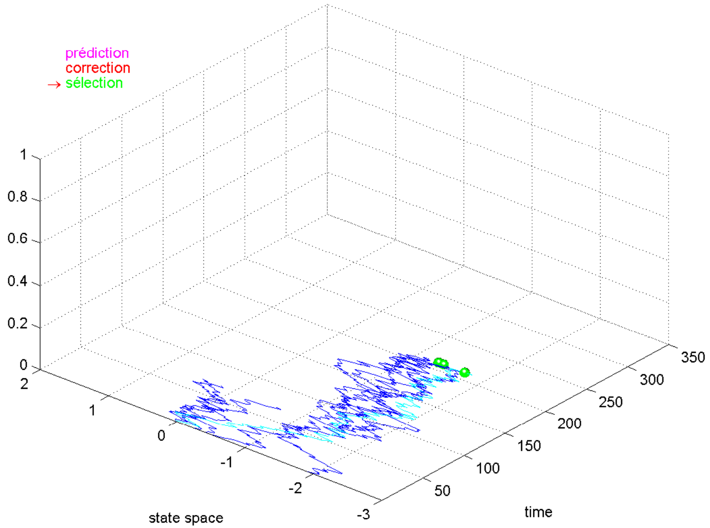
SIS with multinomial selection



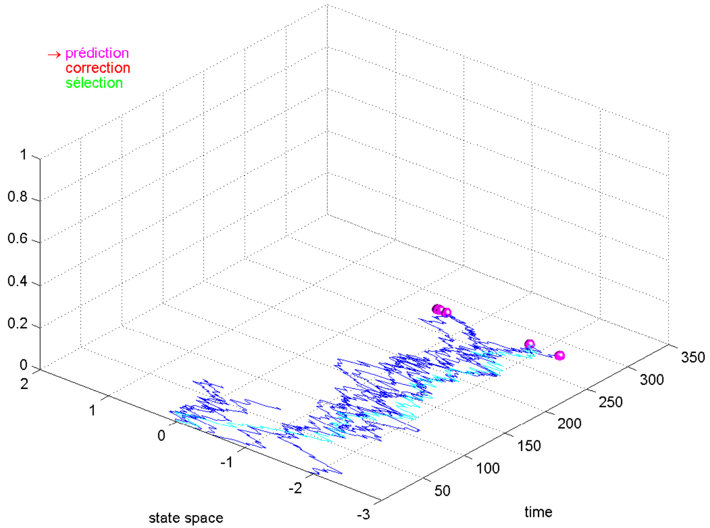
SIS with multinomial selection



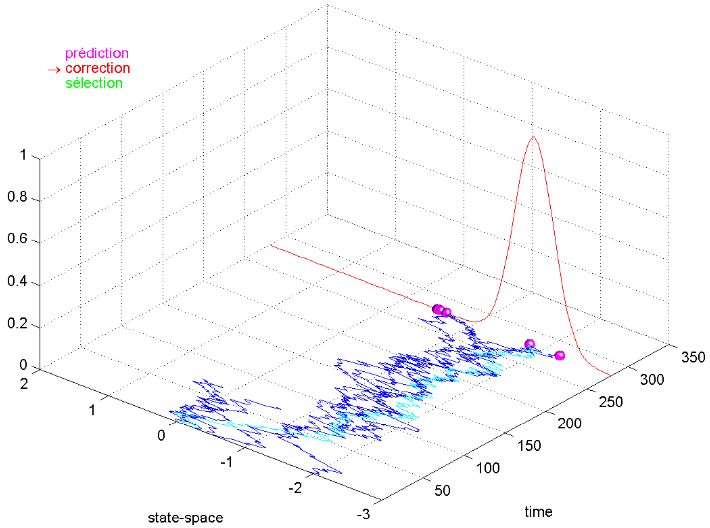
SIS with multinomial selection



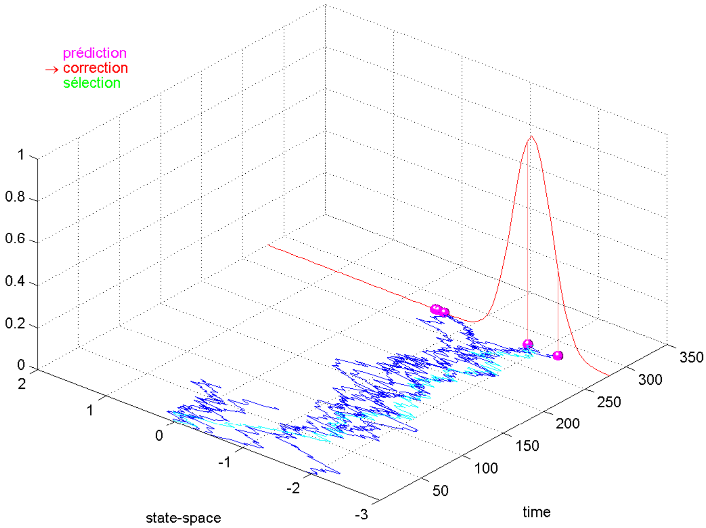
SIS with multinomial selection



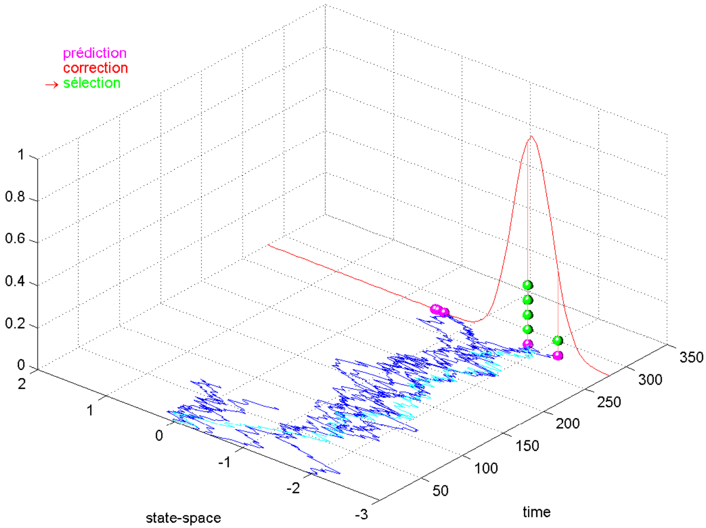
SIS with multinomial selection



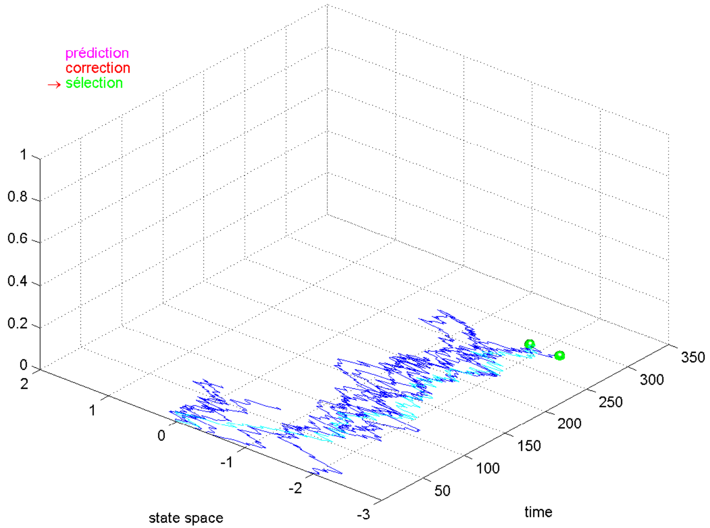
SIS with multinomial selection



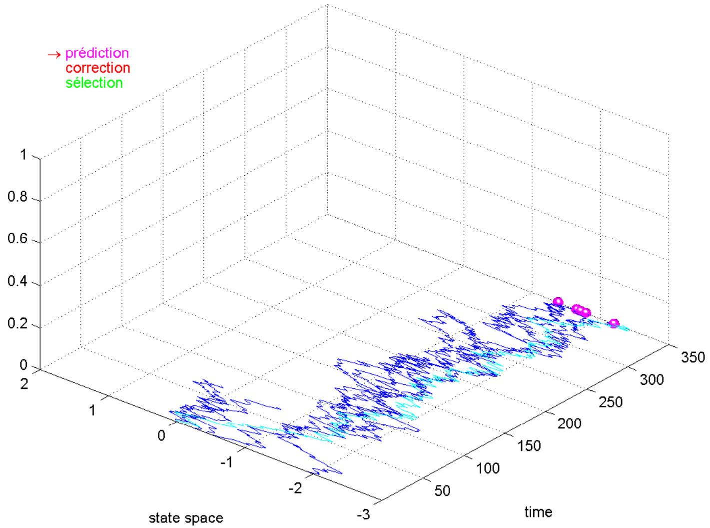
SIS with multinomial selection



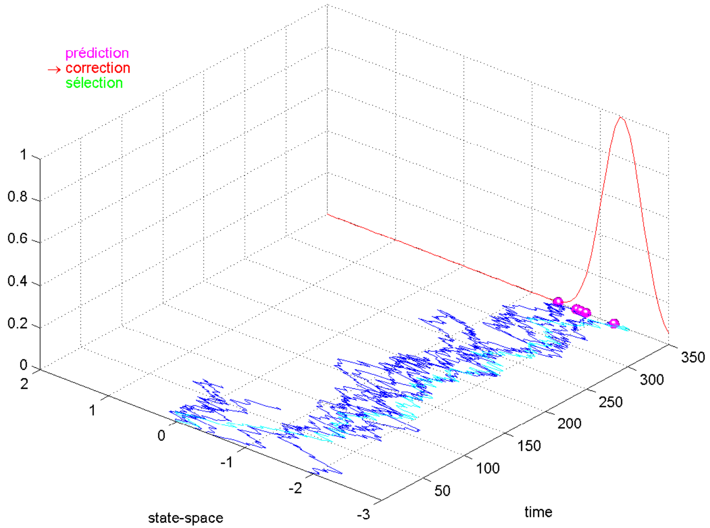
SIS with multinomial selection



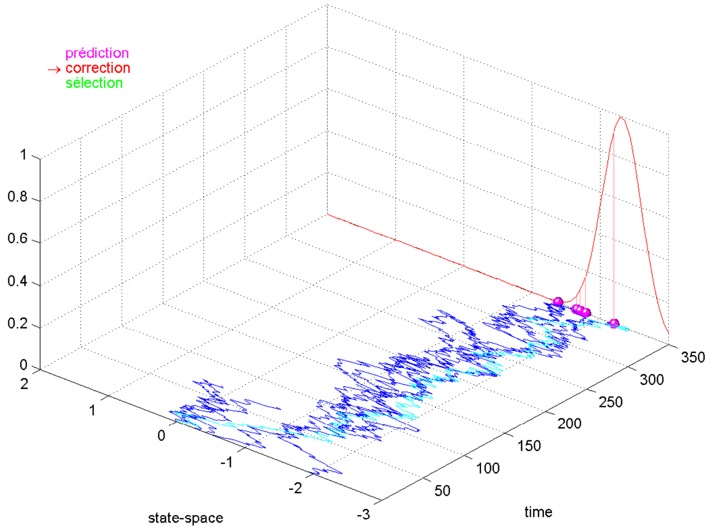
SIS with multinomial selection



SIS with multinomial selection



SIS with multinomial selection



Outline

Residual resampling

- There are several alternatives to multinomial selection. In the residual resampling scheme the number N_n^i of offspring of particle i is—"semi-deterministically"—set to

$$N_n^i = \left\lfloor N \frac{\omega_n^i}{\sum_{\ell=1}^N \omega_n^\ell} \right\rfloor + \tilde{N}_n^i,$$

where the $\tilde{N}_n^i$'s are random integers obtained by randomly distributing the remaining $N - R$ offspring, with

$$R \stackrel{\text{def}}{=} \sum_{j=1}^N \left\lfloor N \frac{\omega_n^j}{\sum_{\ell=1}^N \omega_n^\ell} \right\rfloor,$$

among the ancestors as follows.

Residual resampling, pseudo-code

for $i = 1 \to N$ **do**

  set $\tilde{N}_n^i \leftarrow 0$;

  set $\bar{\omega}_n^i \leftarrow \frac{1}{N-R}\left(N\frac{\omega_n^i}{\sum_{\ell=1}^N \omega_n^\ell} - \left\lfloor N\frac{\omega_n^i}{\sum_{\ell=1}^N \omega_n^\ell}\right\rfloor\right)$;

end

for $r = 1 \to N - R$ **do**

  set $I_r \leftarrow j$ with probability $\bar{\omega}_n^j$;

  set $\tilde{N}_n^{I_r} \leftarrow \tilde{N}_n^{I_r} + 1$;

end

return $(\tilde{N}_n^i)$

Residual resampling, unbiasedness

- Consequently, residual resampling replaces

$$\sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} \phi(X_{0:n}^i) \quad \text{by} \quad \frac{1}{N} \sum_{i=1}^{N} N_n^i \phi(X_{0:n}^i).$$

- Also residual resampling is unbiased (exercise!):

Theorem

For all $N \geq 1$ and $n \geq 0$,

$$\mathbb{E}\left(\frac{1}{N} \sum_{i=1}^{N} N_n^i \phi(X_{0:n}^i) \right) = \mathbb{E}\left(\sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} \phi(X_{0:n}^i) \right).$$

- Residual selection can be proven to have lower variance than multinomial selection.

Other selection strategies

- Other selection strategies are
  - Stratified resampling
  - Bernoulli branching
  - Poisson branching
  - ...

Outline

1. Last time: sequential importance sampling (SIS)
2. IS with resampling (ISR)
3. SIS with resampling (SISR)
   - SIS with multinomial selection
   - Alternative selection strategies
   - Further properties of SISR
4. HA1

SISR: Some theoretical results

- Even though the theory of SISR is hard, there is a number of results establishing the convergence of the algorithm as N tends to infinity. For instance,

$$\sqrt{N}\left(\sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} \phi(X_{0:n}^i) - \tau_n\right) \xrightarrow{\text{d.}} \mathrm{N}(0, \sigma_n^2(\phi)),$$

  where $\sigma_n^2(\phi)$ lacks generally closed-form expression. Note that the convergence rate is still $\sqrt{N}$.
- Open problem: find an effective and numerically stable online estimator of $\sigma_n^2(\phi)$!

SISR: long-term numerical stability

- The dependence of $\sigma_n^2(\phi)$ on n is crucial. However, for filtering in HMMs one may prove, under weak assumptions, that there is a constant $c < \infty$ such that

$$\sigma_n^2 \le c, \quad \forall n \ge 0.$$

  Thus, the SISR estimates are, in agreement with empirical observations, indeed numerically stable in n.
- Otherwise the method would be practically useless!

SISR: particle path degeneracy

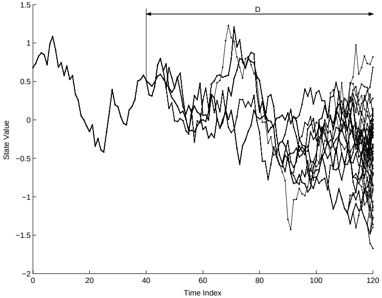


Figure: When $m \ll n$, the set $(X^i_{0:n}(m))_{i=1}^N$ is composed of a few different elements only, yielding poor estimation of the marginal of f_n w.r.t. x_m. However, estimation w.r.t. x_n is always efficient.

SISR: particle path degeneracy (cont.)

- As the particle smoother resamples systematically the particles, it will always hold that for a fixed $k \leq n$,

$$(X^i_{0:n+1}(m))^N_{i=1} \subseteq (X^i_{0:n}(m))^N_{i=1}.$$

- Thus, as time increases, the effective number of (different) particle components $(X^i_{0:n}(m))^N_{i=1}$ at time m will decrease. After while, all particle trajectories have collapsed into a single one. ☹
- Consequently, the estimates will suffer from large variance for large n.
- This is a universal problem when applying the SISR algorithm to the path space.

SISR: particle path degeneracy (cont.)

- Still, in its most basic form, SISR works well
  - in the important case where ϕ is a function of the last component x_n only, i.e.,

    $$\tau_n = \int_{\mathsf{X}_n} \phi(x_n) f_n(x_{0:n}) \, dx_{0:n},$$

    which is the case for, e.g., filtering in HMMs.
  - for estimation of the normalizing constant c_n through $c_{N,n}^{\text{SISR}}$.

SISR: particle path degeneracy (cont.)

- The path degeneracy may be kept back somewhat by resampling the particles only when the skewness of the weights, as measured, e.g., by the coefficient of variation

$$\mathrm{CV}_N = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(N \frac{\omega_n^i}{\sum_{\ell=1}^{N} \omega_n^\ell} - 1 \right)^2},$$

exceeds some threshold.

- The CV_N is minimal (zero) when all weights are equal and maximal ($\sqrt{N-1}$) when only one particle has non-zero weight.
- A related criterion is the so-called efficient sample size given by $N/(1 + \mathrm{CV}_N^2)$.

Coping with path degeneracy

- Recently, advanced extensions of the standard SISR algorithm have been proposed in order to cope with the particle path degeneracy.
- Most of these methods, which are beyond the scope of the course, apply backward simulation on the grid of particle locations formed by the SISR algorithm.
- Efficient online implementation is generally possible in the important case of additive objective functions of form

$$\phi(x_{0:n}) = \sum_{\ell=0}^{n-1} \phi_\ell(x_{\ell:\ell+1}).$$

A few references on particle filtering

- Cappé, O., Moulines, E., and Rydén, T. (2005) *Inference in Hidden Markov Models*. Springer.
- Doucet, A., De Freitas, N., and Gordon, N. (2001) *Sequential Monte Carlo Methods in Practice*. Springer.
- Fearnhead, P. (1998) *Sequential Monte Carlo Methods in Filter Theory*. Ph.D. thesis, University of Oxford.

Outline

HA1: sequential Monte Carlo methods

- HA1 deals with SMC-based mobility tracking in cellular networks. The model is described by an HMM, and the assignment deals with
  - SIS-based optimal filtering of a target's position,
  - an SISR-based approach to the same problem,
  - model calibration via SMC-based maximum likelihood estimation.
- The class E3 after easter will be used as question time for HA1 (or other questions related to the course). Bring your laptop if you want!

Submission of HA1

The following is to be submitted:

- An email containing the report files as well as *all* your m-files with a file `group number-HA1-matlab.m` that runs your analysis. This email has to be sent to `johawes@kth.se` by Friday 28 April, 13:00:00.
- A report, named `group number-HA1-report.pdf`, in PDF format (No MS Word-files) of maximum 7 pages (with names). **One** printed and stitched copy of the report are brought (physically!) to the lecture on Friday 28 April.
- A report, named `HA1-report.pdf`, identical to the above report but without group number or names. This is used for the peer-review which will be sent out through email.