UiO : Matematisk institutt

Det matematisk-naturvitenskapelige fakultet

STK-4051/9051 Computational Statistics Spring 2021
Chapter 6

Instructor: Odd Kolbjørnsen, oddkol@math.uio.no

Problem with presented methods in high dimensions, Rejection sampling

- Assume you want to sample iid $X_i, i = 1, \ldots, n$
- Two methods with rejection sampling $\frac{f(x_i)}{g(x_i)} \leq \alpha^{-1}$
  1. Sample $X_i, i = 1, \ldots, n$ independently.
  2. Sample $\{X_1, X_2, \ldots, X_n\}$ simultaniously
- Expected time to sample complete
  1. $n \cdot \alpha^{-1}$ (nice ☺)
  2. $1 \cdot \alpha^{-n}$ (curse of dimensionality is back 😐)
- For complex distributions in high dimensions we need a joint proposal (within this framework)

Factoring into 1D distributions

- Target
  $f(\boldsymbol{x}) = f(x_1)f(x_2|x_1)\cdots f(x_n|x_1, x_2, \dots, x_{n-1})$
- Proposal
  $g(\boldsymbol{x}) = g(x_1)g(x_2|x_1)\cdots g(x_n|x_1, x_2, \dots, x_{n-1})$
- Sample each component sequentially as 1D
- Markov property: (Simpler notation)
- $f(\boldsymbol{x}) = f(x_1)f(x_2|x_1)f(x_3|x_2)\cdots f(x_n|x_{n-1})$
- $g(\boldsymbol{x}) = g(x_1)g(x_2|x_1)g(x_3|x_2)\cdots g(x_n|x_{n-1})$

Sequential Monte Carlo in a Markov structure

- Sequential Monte Carlo: First high-dimensional setting
- Assume now $\mathbf{x} = \mathbf{x}_{1:t} = (x_1, \ldots, x_t)$ have a Markov structure

$$f_t(\mathbf{x}_{1:t}) = f_1(x_1) \prod_{i=2}^{t} f_i(x_i|x_{i-1})$$

- Also assume a proposal distribution with Markov property:

$$g_t(\mathbf{x}_{1:t}) = g_1(x_1) \prod_{i=2}^{t} g_i(x_i|x_{i-1})$$

- Importance weights:

$$w(\mathbf{x}_{1:t}) = \frac{f_t(\mathbf{x}_{1:t})}{g_t(\mathbf{x}_{1:t})} = \frac{f_1(x_1)}{g_1(x_1)} \prod_{i=2}^{t} \frac{f_i(x_i|x_{i-1})}{g_i(x_i|x_{i-1}))} = w(\mathbf{x}_{1:t-1})\frac{f_t(x_t|x_{t-1})}{g_t(x_t|x_{t-1}))}$$

- Opens up for sequential sampling/estimation
- Note: Easy to generalize to non-Markov settings as well
  - More computing at each step

Sequential Monte Carlo

Algorithm

1. Sample $X_1 \sim g_1(\cdot)$. Let $w_1 = u_1 = f_1(x_1)/g_1(x_1)$. Set $t = 2$
2. Sample $X_t|x_{t-1} \sim g_t(x_t|x_{t-1})$.
3. Append x_t to $\mathbf{x}_{1:t-1}$, obtaining $\mathbf{x}_t$
4. Let $u_t = f_t(x_t|x_{t-1})/g_t(x_t|x_{t-1})$
5. Let $w_t = w_{t-1}u_t$, the importance weight for $\mathbf{x}_{1:t}$
6. Increment t and return to step 2

Can simulate m sequences in paralell!

Sequential Monte Carlo Example

- Assume Markov model

$$X_1 \sim N(0, \sqrt{2})$$
$$f_t(x_t|x_{t-1}) \propto |\cos(X_t - X_{t-1})| \exp\left\{-\tfrac{1}{4}(X_t - X_{t-1})^2\right\}$$

- Of interest:

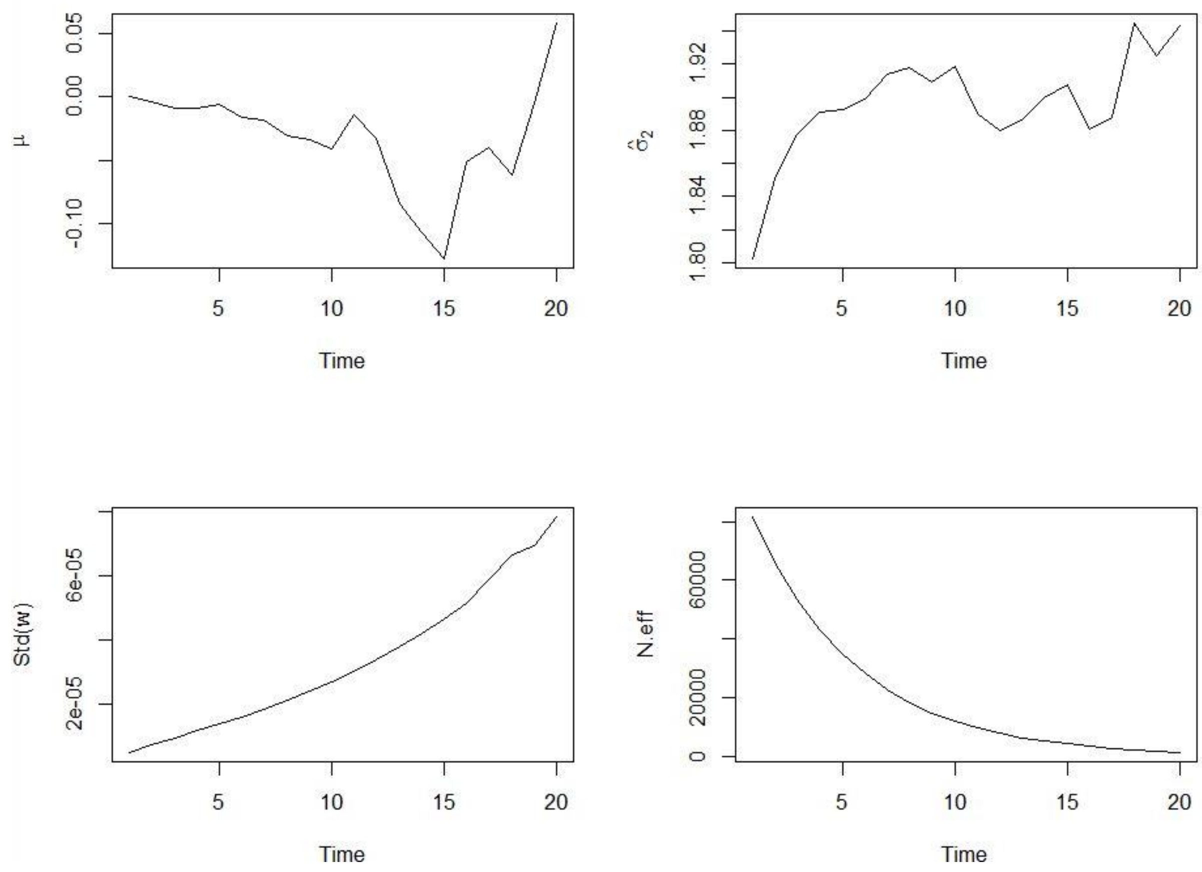
$$\mu_t = E[X_t] \qquad = 0 \text{ due to symmetry}$$
$$\sigma_t^2 = \text{var}(X_t) = E[X_t^2] - (E[X_t])^2$$

- Estimators by SMC simulation:

$$\hat{\mu}_t = \frac{\sum_{i=1}^n w_t^i x_t^i}{\sum_{i=1}^n w_t^i}$$
$$\hat{\sigma}_t^2 = \frac{\sum_{i=1}^n w_t^i (x_t^i)^2}{\sum_{i=1}^n w_t^i} - \left[\frac{\sum_{i=1}^n w_t^i x_t^i}{\sum_{i=1}^n w_t^i}\right]^2 = \frac{\sum_{i=1}^n w_t^i (x_t^i - \hat{\mu}_t)^2}{\sum_{i=1}^n w_t^i}$$

- `SMC_cosnorm.R`

Dimension =20, m=100 000

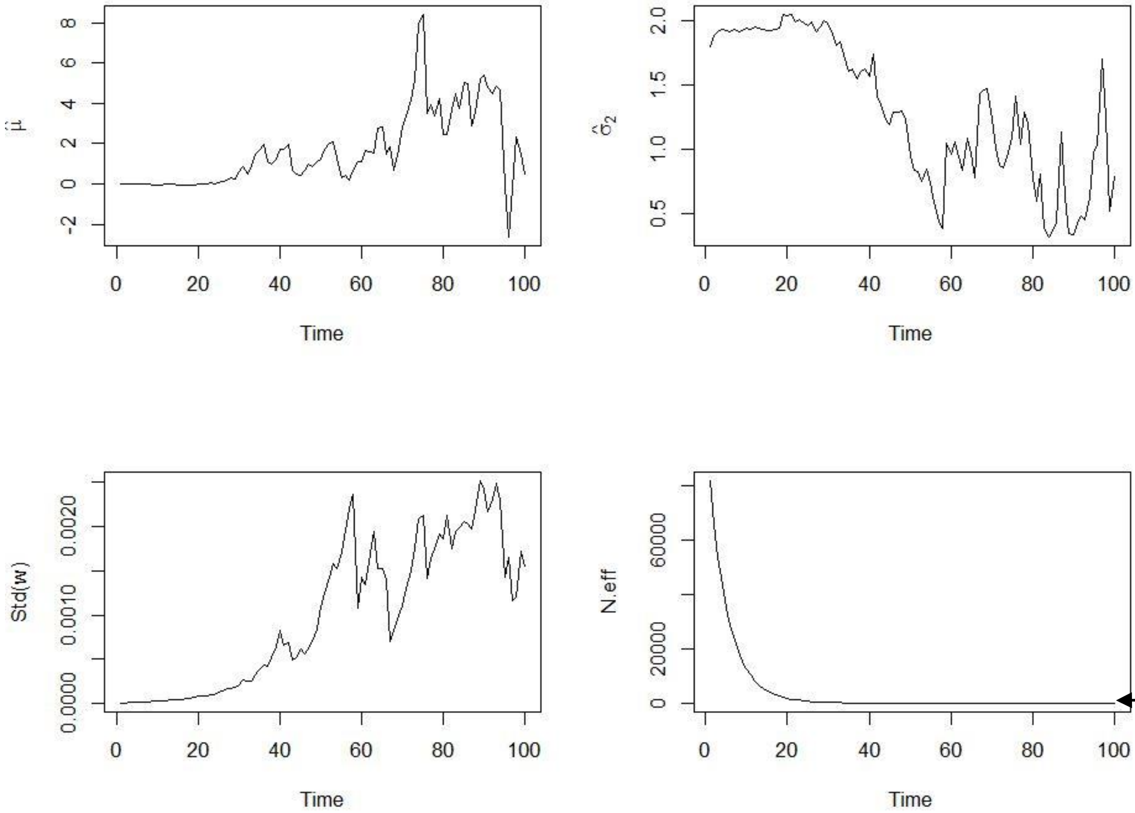


$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{n} w_i^2}$$

Neff: 1514

```
show(w.norm.sort[1:10])
[1] 0.007377638 0.006993864 0.004586039 0.003991772 0.003901192 0.003686038 0.003401063 0.003339764 0.003094683 0.002911
```

Dimension =100, m=100 000



$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{n} w_i^2}$$

Neff: 3.78

```
show(w.norm.sort[1:10])
[1] 0.47751939 0.06007770 0.04960048 0.03573471 0.03281825 0.02935132 0.02665531 0.02271621 0.02231759 0.02197812
```

Weight degeneracy

- General rule:

$$\text{var}[Y] = E[\text{var}[Y|Z]] + \text{var}[E[Y|Z]] \geq \text{var}[E[Y|Z]]$$

- $Y = w_t, Z = \mathbf{X}_{1:t-1}$ (w_{t-1} given by $\mathbf{x}_{1:t-1}$):

$$\begin{aligned} E[w_t|X_{1:t-1}] =& w_{t-1} E[\tfrac{f_t(X_t|X_{t-1})}{g_t(X_t|X_{t-1})}|\mathbf{X}_{1:t-1}] \\ =& w_{t-1} \cdot 1 = w_{t-1} \end{aligned}$$

  implying that

$$\text{var}[w_t] \geq \text{var}[w_{t-1}]$$

  which indicates that the variance will increase at each time-step.
- Practical consequence:
  - Only a few samples will dominate the others
  - Variability of estimate will increase

Resampling

- Degeneracy of weights a serious problem.
- Solution: Resampling (SIR idea)

- One possible choice for resampling: Apply SIR directly

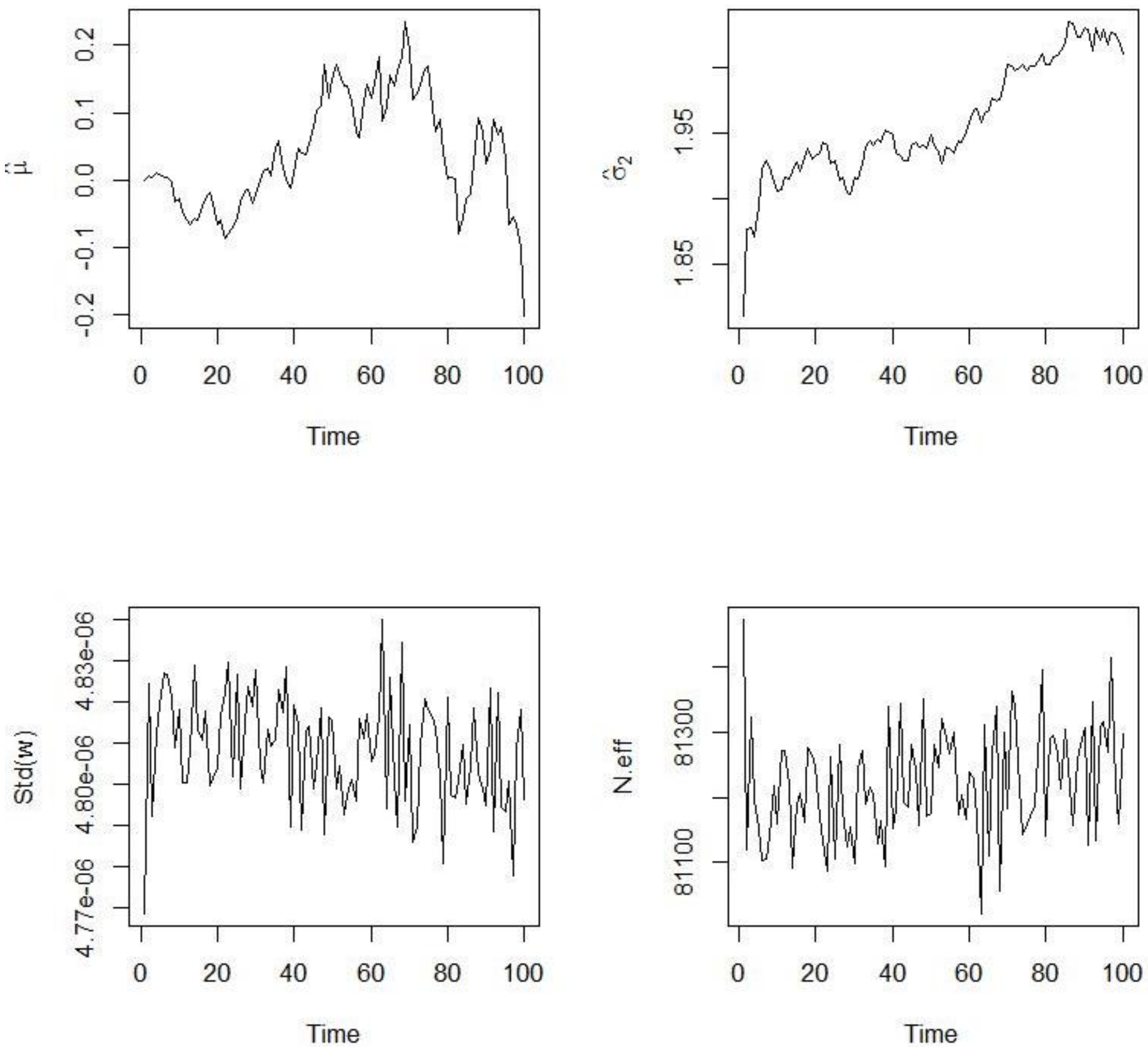
- Each iteration:
  - Sample $\tilde{x}_t^i$, independently from $\{x_t^i\}$, probability of each sample is w_t^i
  - Set all new weigths equal to 1/N

- When N_{eff} is small
  - Sample $\tilde{x}_t^i$, independently from $\{x_t^i\}$, probability of each sample is w_t^i
  - Set all new weigths equal to 1/N

- `SMC_cosnorm.R`

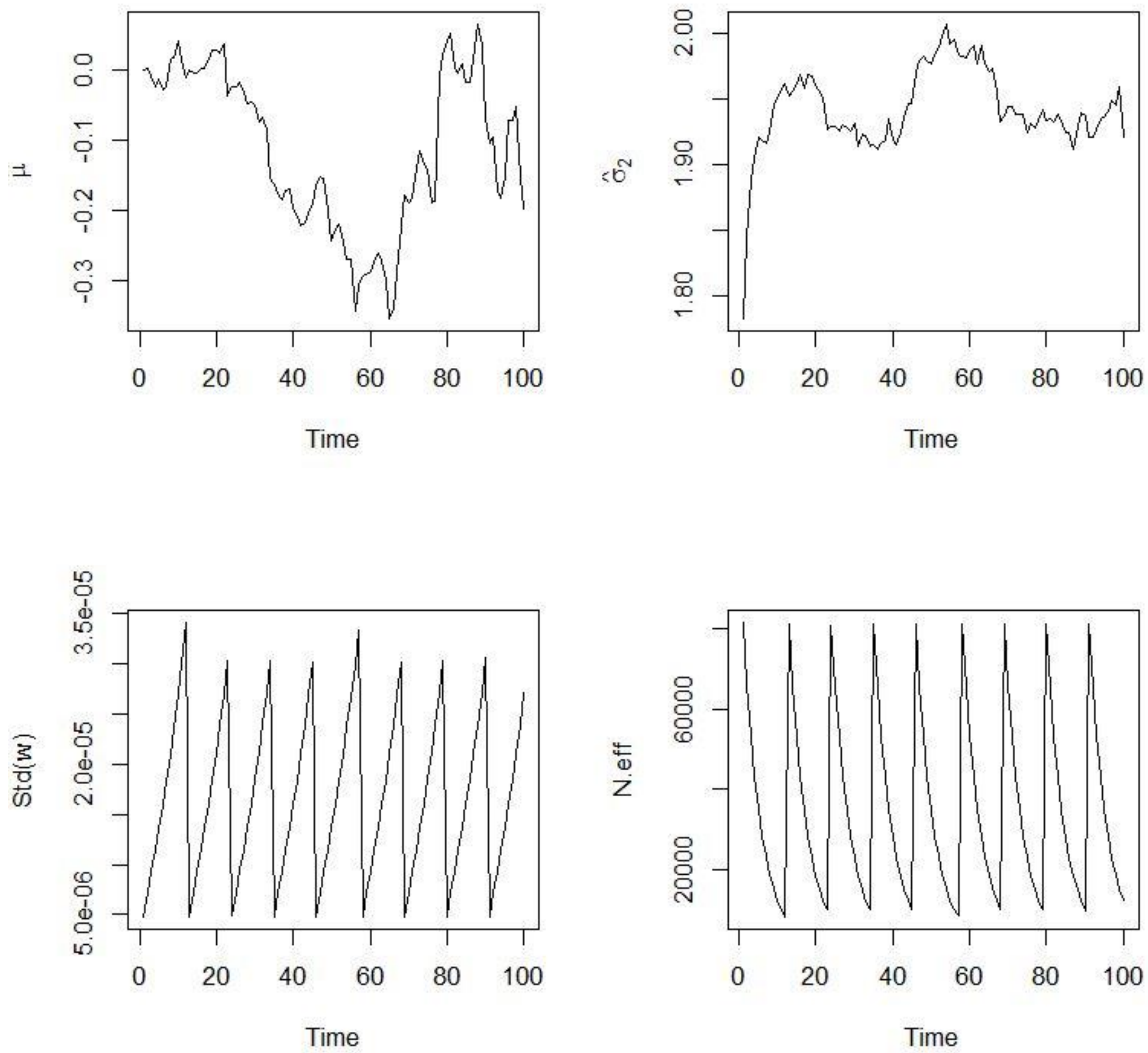
Dimension =100, m=100 000

Resample
each time



Dimension =100, m=100 000

Resample
n<10 000



How to resample more generally?

- Resampling $\tilde{x}_t^i$ from x_t^i :
  - Original (normalized) weights: w_t^i
  - Resampling weights: $\widetilde{w_t}^i$
  - The number of repeats for x_t^i: N_t^i
- We need the expected number of resamples times the new weight to be the old weight
  - $E\left(N_t^i \cdot \widetilde{w_t}^i\right) = w_t^i$
- One choice (as above)
  - Sample $\tilde{x}_t^i$, independently from $\{x_t^i\}$, probability of each sample is w_t^i
  - Set new weights 1/N

Resampling

Original (normalized) weights: w_t^i
Resampling weights: $\widetilde{w_t}^i$
The number of repeats for x_t^i: N_t^i

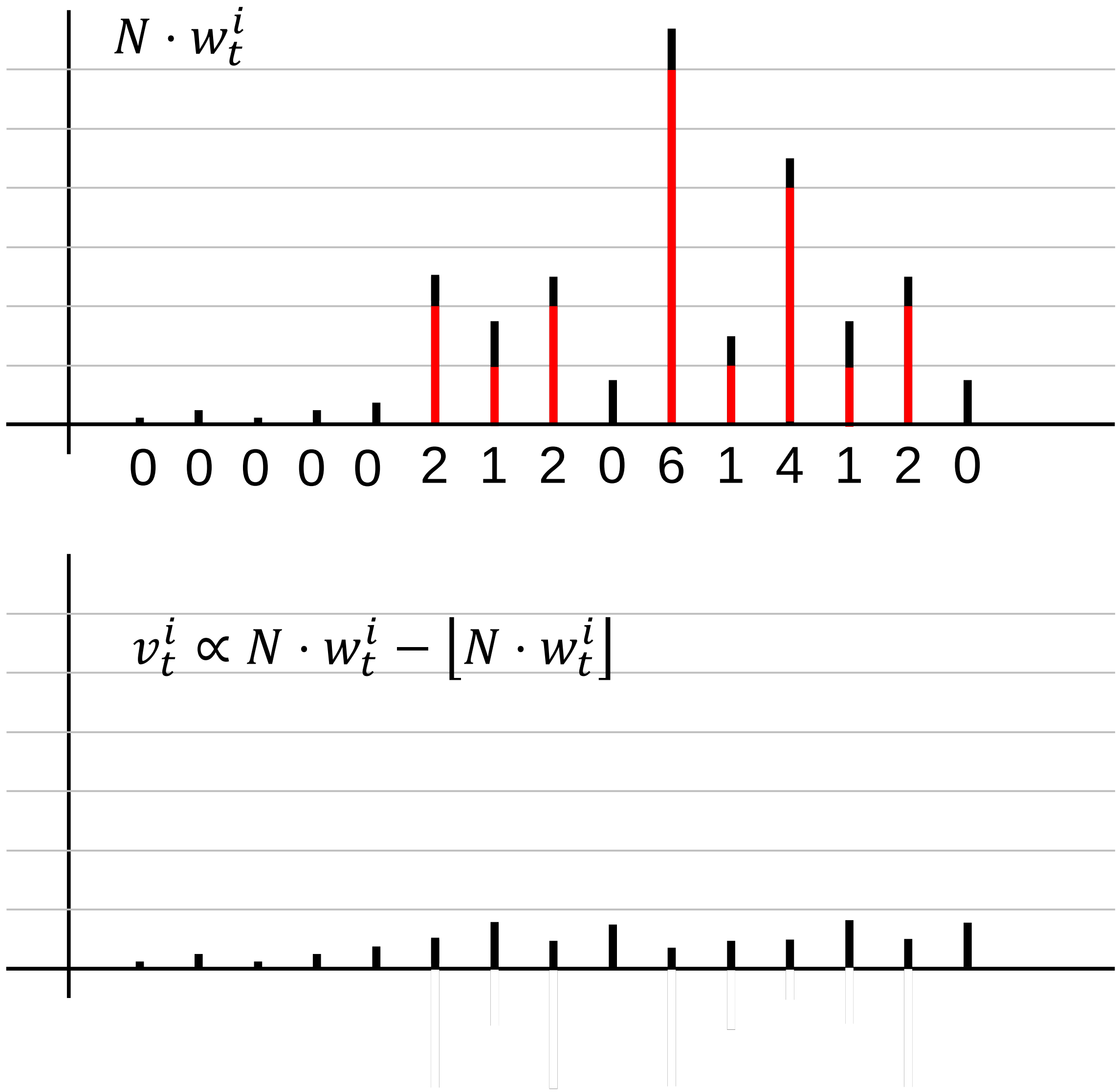
- Simplest option:
  - Resample with probabilities equal to w_t^i.
  - Put weights on resample to $\tilde{w}_t^i = N^{-1}$
  - Number of repeats of x_t^i, N_t^i is Binomial(N, w_t^i)
  - $E[N_t^i \tilde{w}_t^i] = N w_t^i$
- More general resampling strategies are possible
- Sufficient requirement: $E[N_t^i \tilde{w}_t^i] = N w_t^i$
- Optimal strategy (for equally weighted samples)
  - For $i = 1, \ldots, N$, put ($\lfloor a \rfloor$ is the largest integer smaller than a)

    $$\widetilde{N}_t^i = \lfloor N w_t^i \rfloor \quad \text{(Some will be zero)}$$

  - Let $\delta_t^i = w_t^i - \widetilde{N}_t^i / N$
  - Define $K = N - \sum_{i=1}^{N} \widetilde{N}_t^i$ (remaining particles that have not been allocated)
  - Sample $(D_t^1, \ldots, D_t^N)$ from the multinomial distribution with probabilities proportional to $(\delta_t^1, \ldots, \delta_t^n)$.
  - Put $N_t^i = \tilde{N}_t^i + D_t^i$
  - Make N_t^i replicates of x_t^i, but all weights to $1/N$

Illustration of optimal resampling

(resample weigth: $1/N$)



N=25

Total=19

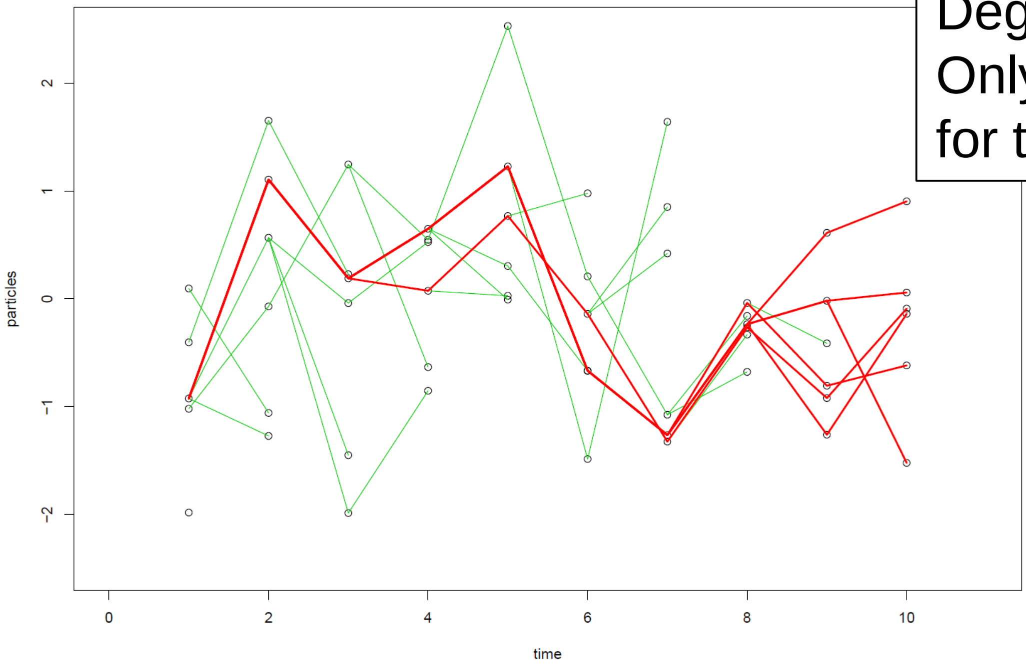
Resample=6

Example optimal resampling

- $N = 5$, $\mathbf{w} = (0.3, 0.4, 0.05, 0.15, 0.2)$
- $N * \mathbf{w} = (1.5, 2.0, 0.25, 0.75, 1.0)$
- $\lfloor N \cdot w_t^i \rfloor$ $\tilde{\mathbf{N}} = (1, 2, 0, 0, 1)$
- $K = 5 - 4 = 1$
- $\tilde{\mathbf{N}}/N = (0.2, 0.4, 0.0, 0.0, 0.2)$
- $\delta = (0.1, 0.0, 0.05, 0.15, 0.0)$
- Sample $\mathbf{D}$ from Multinom$(1 : N, 1, (\frac{0.1}{0.3}, \frac{0.0}{0.3}, \frac{0.05}{0.3}, \frac{0.15}{0.3}, \frac{0.0}{0.3}))$ e.g $\mathbf{D} = (1, 0, 0, 0, 0)$
- Put $\mathbf{N} = \tilde{\mathbf{N}} + \mathbf{D} = (2, 2, 0, 0, 1)$

Resampling properties

- Resampling will introduce extra random noise at the current time-point
- Can reduce noise at later time points
- Gives a good approximation to $f(x_n)$



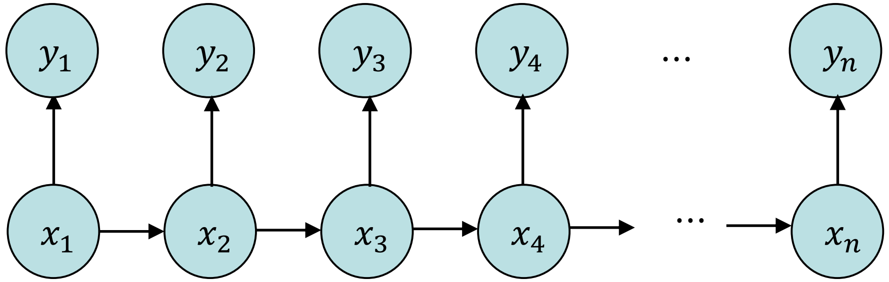
Degenerated sample path
Only one particle left
for t=1,2 and 3

- Does not give a good approximation to $f(\mathbf{x})$ or $f(x_1)$!

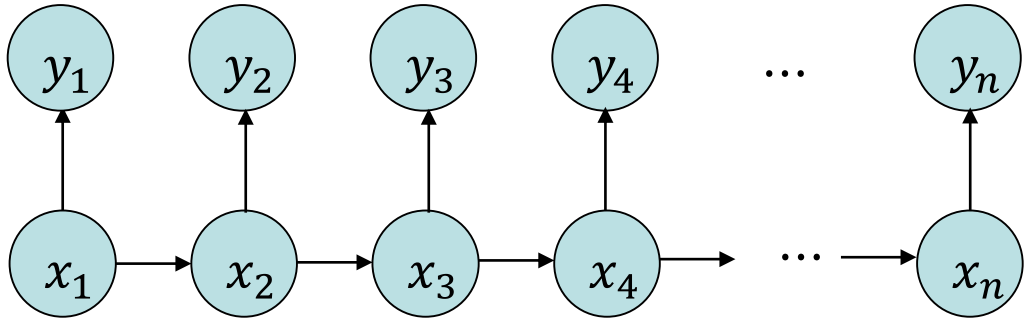
Resampling properties

- Resampling will introduce extra random noise at the current time-point
- Can reduce noise at later time points
- Gives a good approximation to $f(x_n)$
- Does not give a good approximation to $f(\mathbf{x})$ or $f(x_1)$!

Makes it suited for filter problem $p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n})$



Origin of method in state space models

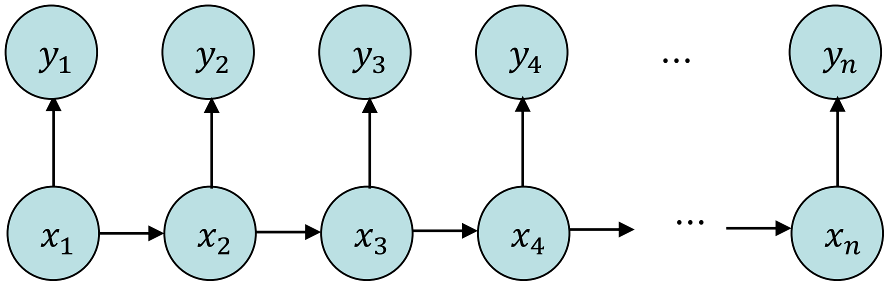


- For the filtering problem the target is the distribution $p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n})$
- In chapter 4, we consider a hidden Markov model the distribution of x (state) is discrete here x (the state) it can be continuous

Hidden Markov models - state space models

- Assume

$$X_1 \sim p_{x_1}(x_1)$$
$$X_t \sim p_x(x_t|x_{t-1})$$
$$Y_t \sim p_y(y_t|x_t)$$



- $\{y_t\}$ observed, $\{x_t\}$ hidden
- Chapter 4: $\{x_t\}$ discrete. Now possibly continuous
- Aim: $f(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})$ or $f(\mathbf{x}_t|\mathbf{y}_{1:t})$
- Recursive relationship (misprint in book):

$$f(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) = \frac{f(\mathbf{x}_{1:t}, y_t|\mathbf{y}_{1:t-1})}{f(y_t|\mathbf{y}_{1:t-1})}$$
$$= \frac{f(\mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1})p_x(x_t|x_{t-1})p_y(y_t|x_t)}{f(y_t|\mathbf{y}_{1:t-1})}$$
$$\propto f(\mathbf{x}_{1:t-1}|y_{1:t-1})p_x(x_t|x_{t-1})p_y(y_t|x_t)$$

Sequential Monte Carlo and HMM

- Assume $g_t(x_t|x_{t-1}) = p_x(x_t|x_{t-1})$

$$\begin{aligned}
w_t =& \frac{f(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})}{g(\mathbf{x}_{1:t})} \\
\propto& \frac{f(\mathbf{x}_{1:t-1}|y_{1:t-1})p_x(x_t|x_{t-1})p_y(y_t|x_t)}{p_{x_1}(x_1)\prod_{s=2}^t p_x(x_s|x_{s-1})} \\
=& \frac{f(\mathbf{x}_{1:t-1}|y_{1:t-1})}{g(\mathbf{x}_{1:t-1})}\frac{p_x(x_t|x_{t-1})p_y(y_t|x_t)}{p_x(x_t|x_{t-1})} \\
=& w_{t-1}p_y(y_t|x_t)
\end{aligned}$$

- Algorithm
  1. Sample $X_1^i \sim p_{x_1}(\cdot), i = 1, ..., n.$
  2. Let $w_1^{*i} = u_1^i = p_y(y_1|x_1^i)$, normalize to $w_i^i = w_1^{*i} / \sum_j w_1^{*j}$. Set $t = 2$
  3. Sample $X_t^i|x_{t-1}^i \sim p_x(x_t|x_{t-1}^i), i = 1, ..., n.$
  4. Append x_t^i to $\mathbf{x}_{1:t-1}^i$, obtaining $\mathbf{x}_t^i$
  5. Let $u_t^i = p_y(y_t|x_t^i)$
  6. Let $w_t^{*i} = w_{t-1}^i u_t^i$, normalize to $w_t^i = w_t^{*i} / \sum_j w_t^{*j}$.
  7. If $\hat{N}_{eff}$ is small, perform resampling
  8. Increment t and return to step 3

Terrain navigation (next time)

- Assume movement model for airplane

$$
\begin{aligned}
\mathbf{X}_t =& \mathbf{x}_{t-1} + \mathbf{d}_t + \boldsymbol{\varepsilon}_t \\
\mathbf{d}_t =& \text{Drift of plane measured by internal} \\
& \text{navigation system (assumed known)} \\
\boldsymbol{\varepsilon}_t =& \mathbf{R}_t^T \mathbf{Z}_t \\
\mathbf{R}_t =& \frac{1}{x_{1,t-1}^2 + x_{2,t-1}^2} \begin{pmatrix} -x_{1,t-1} & x_{2,t-1} \\ -x_{2,t-1} & -x_{1,t-1} \end{pmatrix} \\
\mathbf{Z}_t \sim& N_2\left(\mathbf{0}, q^2 \begin{pmatrix} 1 & 0 \\ 0 & k^2 \end{pmatrix}\right) \qquad q = 400, k = 0.5 \\
Y_t =& m(\mathbf{x}_t) + \delta_t \\
m(\mathbf{x}_t) =& \text{Elevation at point } \mathbf{x}_t
\end{aligned}
$$

- `Example_6_7.R`