Remark 8.14 (*). In self-normalised IS, we have *almost sure* convergence instead of *in distribution*. We state the results here using the latter, because we regard now Algorithm 8.15 to be run with a fixed n — and therefore 'adding samples' does not make immediate sense, but the algorithm may just be repeated with a higher n...

8.4 The particle filter

Algorithm 8.15 (Particle filter)**.** In each line of the algorithm, $i = 1, \ldots, n$:

(i) Sample $X_1^{(i)} \sim M_1$ and set $\mathbf{X}_1^{(i)} = X_1^{(i)}$.

(ii) Calculate $\omega_1^{(i)} := G_1(\mathbf{X}_1^{(i)})$ and set $\bar\omega_1^{(i)} := \omega_1^{(i)}/\omega_1^*$ where $\omega_1^* = \sum_{j=1}^n \omega_1^{(j)}$.

For $t = 2, \ldots, T$, do:

(iii) Sample $A_{t-1}^{(i)} \sim \text{Categorical}(\bar\omega_{t-1}^{(1:N)})$, that is, $\mathbb{P}(A_{t-1}^{(i)} = j) = \bar\omega_{t-1}^{(j)}$.

(iv) Sample $X_t^{(i)} \sim M_t(\,\cdot \mid \mathbf{X}_{t-1}^{(A_{t-1}^{(i)})})$ and set $\mathbf{X}_t^{(i)} = (\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, X_t^{(i)})$.

(v) Calculate $\omega_t^{(i)} := G_t(\mathbf{X}_t^{(i)})$ and set $\bar\omega_t^{(i)} := \omega_t^{(i)}/\omega_t^*$ where $\omega_t^* = \sum_{j=1}^n \omega_t^{(j)}$.

Report $(V^{(1:n)}, \mathbf{X}^{(1:n)})$ where $V^{(j)} := \big(\prod_{t=1}^T \frac{1}{n}\omega_t^*\big)\bar\omega_T^{(j)}$ and $\mathbf{X}^{(j)} := \mathbf{X}_T^{(j)}$.
(In case $\omega_t^* = 0$, the algorithm is terminated with $V^{(i)} = 0$ and with arbitrary $\mathbf{X}^{(i)} \in \mathsf{S}^T$.)

Remark 8.16. The proposal $M_t(x_t \mid x_{1:t-1})$ and the potential $G_t(x_{1:t})$ typically depend on x_t and perhaps x_{t-1}, but not $x_{1:t-2}$. In such a case, it is not necessary to explicitly store $\mathbf{X}_t^{(i)}$, because $\omega_t^{(i)} = G_t(X_{t-1}^{(A_{t-1}^{(i)})}, X_t)$ and $\mathbf{X}^{(i)} = \mathbf{X}_T^{(i)}$ may be recovered from $X_{1:T}^{(j)}$ and $A_{1:T-1}^{(j)}$.

Example 8.17. Implementation with $M_t(x_t \mid x_{1:t-1}) = M_t(x_t \mid x_{t-1})$ and $G_t(x_{1:t}) = G_t(x_t)$:

```
function norm_logw(logw) # Normalised probabilities from log weights ('log-sum-trick')
  m = maximum(logw); u = exp.(logw.-m); return m+log(mean(u)), u/sum(u)
end
function pf(M, logG, n, T)  # Univariate particle filter
    X = zeros(n, T); A = zeros(Int, n, T); wu = zeros(n)
    for i = 1:n
        X[i,1] = x = M(1); wu[i] = logG(1, x)
    end
    V, omega = norm_logw(wu);
    for t = 2:T
        a = rand(Categorical(omega), n); A[:,t-1] = a
        for i = 1:n
            X[i,t] = x = M(t, X[a[i],t-1]); wu[i] = logG(t, x)
        end
        V_, omega = norm_logw(wu); V += V_
  end
  XT = zeros(n,T); XT[:,T]=X[:,T]; a = collect(1:n) # Trace back X^{(i)}
  for t = T-1:-1:1  a = A[a,t]; XT[:,t] = X[a,t]  end
  (logV=V.+log.(omega), XT=XT, X=X)
end
```

Application in Example 8.6, with an estimate for $\mathbb{E}_p[X]$:

```julia
using Distributions, Random, Plots
Random.seed!(42); T=50; x0=0; rho=sigma_x=sigma_y=1
function M(t, x=0.0) # Generate observations from M_t(.|x)
    rand(Normal(x, sigma_x))
end
x_true = zeros(T); x_true[1] = M(1)           # Generate synthetic data:
for t = 2:T x_true[t] = M(t, x_true[t-1]) end # trajectory of x_{1:T}
y = x_true + rand(Normal(0, sigma_y), T)      # and corresponding observations
function logG(t, x) # Calculate log G_t(x)
    logpdf(Normal(y[t], sigma_y), x)
end
o = pf(M, logG, 100, T)
scatter(o.X', color=:black); plot!(o.XT', width=2, legend=false)
sum(norm_logw(o.logV)[2] .* o.XT[:,T])
```

Under certain technical assumptions [cf. 7]:

$$\mathrm{PF}^{(n)}_{M_{1:T},G_{1:T}}(f) := \frac{\sum_{k=1}^n V^{(k)} f(\mathbf{X}^{(k)})}{\sum_{j=1}^n V^{(j)}} = \sum_{k=1}^n \bar{\omega}_T^{(i)} f(\mathbf{X}_T^{(k)}) \xrightarrow{n\to\infty} \mathbb{E}_p[f(X_{1:T})], \quad (23)$$

in distribution.

Remark 8.18. While (23) holds quite generally, the estimator $\mathrm{PF}^{(n)}_{M_{1:T},G_{1:T}}(f)$ typically converges

- quickly for functions that depend only on the last variable (or few last variables), that is, $f(x_{1:T}) = f(x_T)$ (or $f(x_{1:T}) = f(x_{(T-l):T})$ for $l \ll T$). [In the PF, the 'intermediate' distributions π_t are called the *filtering* distributions, from which the name *particle filter* arises.]
- much slower for $f(x_{1:T}) = f(x_1)$ when T is large.

In the latter case, instead of increasing n in a single run of PF, the algorithm may be run several times with fixed n...

Remark 8.19 (*). The step (iii) in Algorithm 8.15 is called *resampling* or *selection*. Algorithm 8.15 was introduced for SSMs in [10], using the specific choice $M_t = m_t$; this algorithm is known as the *bootstrap filter*. The rationale of resampling is, in intuitive terms, to discard 'unlikely paths', and concentrate on 'good candidates.' Similar procedure is used also in genetic algorithms, which aim for (global) optimisation.

Remark 8.20 (*). In fact, the *multinomial resampling* (iii) may be replaced with another procedure drawing non-independent set of indices $A_{t-1}^{(1:n)}$, which still satisfy *unbiasedness*, in the following sense:

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \mathbf{1}\left(A_{t-1}^{(i)} = j\right) \,\middle|\, X_1^{(1:n)}, X_{t-1}^{(1:n)}, A_1^{(1:n)}, \ldots, A_{t-2}^{(1:n)}\right] = \bar{\omega}_{t-1}^{(j)}.$$

For instance, stratified sampling is commonly used, and other choices are possible [cf. 4]. (NB: Even though stratification makes one-step conditional variance smaller, this does not necessarily mean more efficient overall estimator $\mathrm{PF}^{(n)}_{M_{1:T},G_{1:T}}(f)$, even though this is commonly observed empirically...)

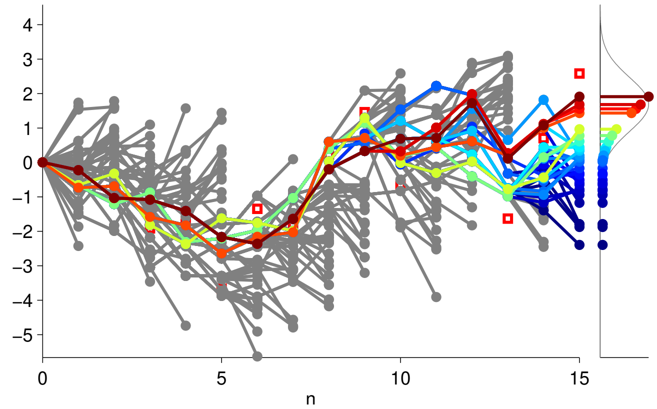
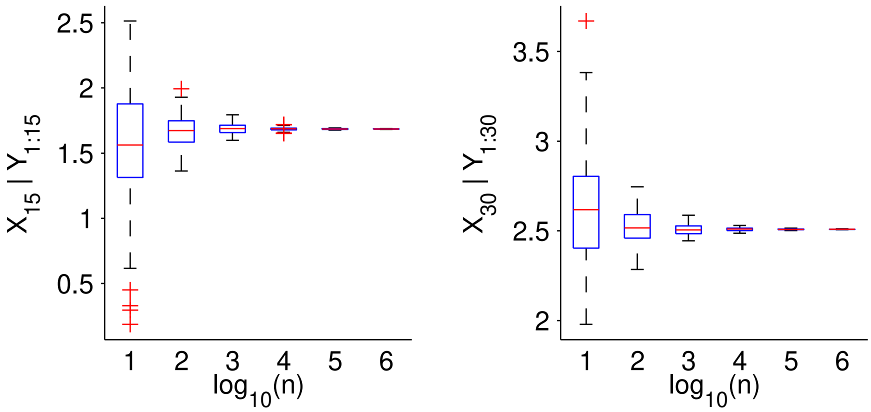


Figure 26: Some samples corresponding to the PF in Example 8.21. The grey paths show the 'dead branches': the ones that were not selected in resampling.



(a) (b)

Figure 27: Box plot of the PF estimates with M_t corresponding to the prior, Example 8.6. Compare with 21. The estimates outperform also SIS with the 'optimal' proposal density in Example 8.7; see Figure 23.

Example 8.21. Let us revisit Example 8.6 with the particle filter; Figure 26 shows the samples produced. It is clear that the resampling helps to concentrate paths (compare with Figure 22). Figure 27 shows a summary of estimates, analogous to Figure 21, and Figure 28 demonstrates that the PF is reliable even for long series of observations, even with this simple proposal distribution.

(Choosing M_t to be q_t as in Example 8.7 would make the results even better, but it is noteworthy that even with $M_t = m_t$, the PF appears to perform reasonably well for bigger T...)

8.5 Unbiasedness of the particle filter

We shall not pursue a detailed proof of (23), but instead focus on the following non-asymptotic unbiasedness property of the PF [cf. 7, Theorem 7.4.2], which turns out to be key property for particle MCMC, which we discuss later.

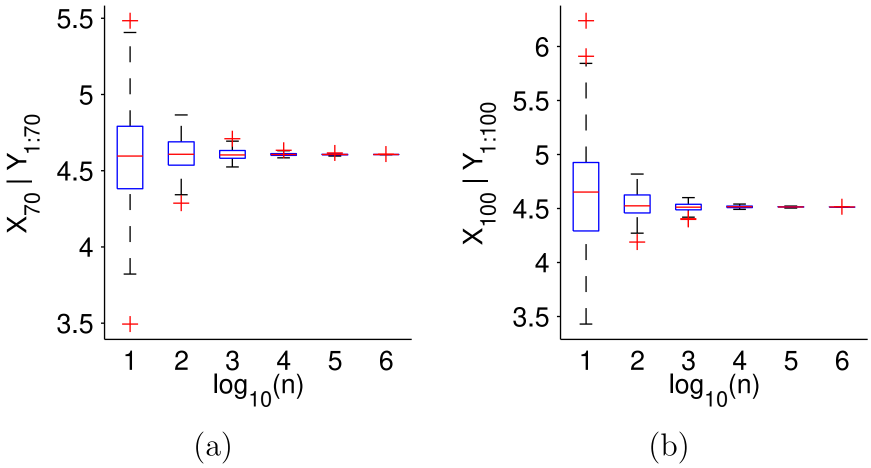


Figure 28: Box plot of the particle filter estimates with M_t corresponding to the prior, Compare with Figure 25. True value for $T = 100$ is approximately 4.514.

Theorem 8.22. *Under assumption* (22)*, for any* $f : \mathbb{S}^T \to \mathbb{R}$ *with* $\mathbb{E}_p[f(X)] < \infty$*, and any* $n \in \mathbb{N}$*, the following holds for the output of Algorithm 8.15:*

$$\mathbb{E}\left[\sum_{k=1}^{n} V^{(k)} f(\mathbf{X}^{(k)})\right] = \int p_u(x_{1:T}) f(x_{1:T}) \mathrm{d}x_{1:T}.$$

Proof. (*) Define the functions $f_T(x_{1:T}) := f(x_{1:T})$, and for $t = T, \ldots, 2$

$$f_{t-1}(x_{1:t-1}) := \int f_t(x_{1:t}) M_t(x_t \mid x_{1:t-1}) G_t(x_{1:t}) \mathrm{d}x_t.$$

Assumption (22) implies that $f_0 := \int M_1(x_1)G_1(x_1)f_1(x_1)\mathrm{d}x_1$ coincides with the desired integral, and all f_t are necessarily (almost everywhere) well-defined if the latter integral is well-defined.

Let us denote $X_{1:t}^{(*)} := \{X_{1:t}^{(i)}, i \in \{1{:}n\}\}$ and similarly $A_{1:t}^{(*)}$, and observe first that for $t = 2, \ldots, T$ and $i \in \{1{:}n\}$,

$$\begin{aligned}
&\mathbb{E}\big[\omega_t^{(i)} f_t(\mathbf{X}_t^{(i)}) \mid X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)}\big] \\
&= \mathbb{E}\Big[\mathbb{E}\big[\omega_t^{(i)} f_t(\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, X_t^{(i)}) \mid X_{1:t-1}^{(*)}, A_{1:t-1}^{(*)}\big] \Big| X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)}\Big] \\
&= \mathbb{E}\bigg[\int M_t(x_t \mid \mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}) G_t(\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, x_t) f_t(\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, x_t) \mathrm{d}x_t \,\bigg|\, X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)}\bigg] \\
&= \sum_{j=1}^{n} \mathbb{P}\big(A_{t-1}^{(i)} = j \mid X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)}\big) f_{t-1}(\mathbf{X}_{t-1}^{(j)}),
\end{aligned}$$

so we may conclude that

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \omega_t^{(i)} f_t(\mathbf{X}_t^{(i)}) \,\middle|\, X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)}\right] = \sum_{j=1}^{n} \bar{\omega}_{t-1}^{(j)} f_{t-1}(\mathbf{X}_{t-1}^{(j)}). \tag{24}$$

We may apply (24) recursively with $t = T, \dots, 2$, yielding

$$\mathbb{E}\Big[\sum_{k=1}^n V^{(k)} f(\mathbf{X}_T^{(k)})\Big] = \mathbb{E}\Big[\Big(\prod_{t=1}^{T-1} \frac{1}{n}\omega_t^*\Big)\mathbb{E}\Big[\Big(\frac{1}{n}\omega_T^*\Big)\sum_{i=1}^n \bar{\omega}_T^{(i)} f_T(\mathbf{X}_T^{(i)}) \;\Big|\; X_{1:T-1}^{(*)}, A_{1:T-2}^{(*)}\Big]\Big]$$

$$= \mathbb{E}\Big[\Big(\prod_{t=1}^{T-1} \frac{1}{n}\omega_t^*\Big)\sum_{i=1}^n \bar{\omega}_{T-1}^{(i)} f_{T-1}(\mathbf{X}_{T-1}^{(i)})\Big] = \dots$$

$$= \frac{1}{n}\sum_{i=1}^n \mathbb{E}\Big[\omega_1^* \bar{\omega}_1^{(i)} f_1(X_1^{(i)})\Big],$$

which equals to f_0 by a similar calculation as above. □

One immediate consequence of the unbiasedness is that we may *combine* easily the output of independent particle filters, and deduce a consistent estimator as in self-normalised IS:

Corollary 8.23 (*)**.** *Fix $n \in \mathbb{N}$ and suppose $(V^{(1:n)}, \mathbf{X}^{(1:n)})$ is the output of Algorithm 8.15. Let $\zeta(f) := \sum_{k=1}^n V^{(k)} f(\mathbf{X}^{(k)})$ and $\zeta(1) := \sum_{k=1}^n V^{(k)}$. Suppose $(\zeta_i(f), \zeta_i(1))_{i\ge 1}$ are independent realisations of $(\zeta(f), \zeta(1))$, then*

(i) $E_{M_{1:T},G_{1:T}}^{(N,n)}(f) := \dfrac{\sum_{i=1}^N \zeta_i(f)}{\sum_{j=1}^N \zeta_j(1)} \xrightarrow{N\to\infty} \mathbb{E}_p[f(X)]$ *a.s.*

(ii) If $\mathbb{E}\big[|\zeta(f) - \zeta(1)\mathbb{E}_p[f(X)]|^2 + |\zeta(1)|^2\big] < \infty$, then for any $\alpha \in (0,\infty)$,

$$\mathbb{P}\Big(\mathbb{E}_p[f(X)] \in \Big[E_{M_{1:T},G_{1:T}}^{(N,n)}(f) \pm \alpha\sqrt{\hat{v}^{(N,n)}}\Big]\Big) \xrightarrow{N\to\infty} 1 - 2\Phi(-\alpha), \quad \textit{where}$$

$$\hat{v}^{(N,n)} := \frac{\sum_{i=1}^N \big(\zeta_i(f) - \zeta_i(1)E_{M_{1:T},G_{1:T}}^{(N,n)}(f)\big)^2}{\big(\sum_{j=1}^N \zeta_k(1)\big)^2}.$$

Proof. (i) follows from Theorem 8.22, because $\mathbb{E}[\zeta(f)]/\mathbb{E}[\zeta(1)] = \mathbb{E}_p[f(X)]$, and (ii) follows similarly as Theorem 4.23, once we observe that as $N \to \infty$,

$$N\hat{v}^{(N,n)} = \frac{\frac{1}{N}\sum_{i=1}^N \big(\zeta_i(f) - \zeta_i(1)E_{M_{1:T},G_{1:T}}^{(N,n)}(f)\big)^2}{\big(\frac{1}{N}\sum_{j=1}^N \zeta_k(1)\big)^2} \to \frac{\mathbb{E}\big[(\zeta(f) - \zeta(1)\mathbb{E}_p[f(X)])^2\big]}{\mathbb{E}_p[\zeta(1)]^2}.$$

□

Remark 8.24 (*). Suppose $\mathrm{PF}_{M_{1:T},G_{1:T}}^{(n,i)}(f)$ are independent realisations of $\mathrm{PF}_{M_{1:T},G_{1:T}}^{(n)}(f)$ in (23), then, unlike $E_{M_{1:T},G_{1:T}}^{(N,n)}(f)$, the naive combination $\frac{1}{N}\sum_{i=1}^N \mathrm{PF}_{M_{1:T},G_{1:T}}^{(n,i)}(f)$ is *not* consistent, because $\mathbb{E}[\mathrm{PF}_{M_{1:T},G_{1:T}}^{(n)}(f)] \neq E_p[f(X)]$ in general (even though, under general conditions, $\mathbb{E}[\mathrm{PF}_{M_{1:T},G_{1:T}}^{(n)}(f)] \to E_p[f(X)]$ as $n \to \infty$). On the contrary, the estimator $E_{M_{1:T},G_{1:T}}^{(N,n)}(f)$ is consistent with any n, and only requires an asymptotic in N.

9 Particle MCMC

As a final topic of the course, we discuss a particle MCMC algorithm introduced in the seminal paper [3]. It is based on a combination of MCMC and particle filter, in a way that allows for Bayesian inference in a parameterised SSM.