Advanced Sequential Monte Carlo Methods

Arnaud Doucet
Departments of Statistics & Computer Science
University of British Columbia

Generic Sequential Monte Carlo Scheme

- At time $n = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$.
- Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$
- At time $n \geq 2$
  - sample $X_n^{(i)} \sim q_n\left(\cdot \middle| X_{1:n-1}^{(i)}\right)$
  - compute $w_n\left(X_{1:n}^{(i)}\right) = \dfrac{\gamma_n\left(X_{1:n}^{(i)}\right)}{\gamma_{n-1}\left(X_{1:n-1}^{(i)}\right) q_n\left(X_n^{(i)} \middle| X_{1:n-1}^{(i)}\right)}$.
- Resample $\left\{X_{1:n}^{(i)}, W_n^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_{1:n}^{(i)}\right\}$

Sequential Monte Carlo for Hidden Markov Models

- At time $n = 1$, sample $X_1^{(i)} \sim q_1(\cdot)$ and set
  $$w_1\left(X_1^{(i)}\right) = \frac{\mu\left(X_1^{(i)}\right) g\left(y_1 | X_1^{(i)}\right)}{q\left(X_1^{(i)} \middle| y_1\right)}.$$
- Resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_1^{(i)}\right\}$
- At time $n \geq 2$
  - sample $X_n^{(i)} \sim q\left(\cdot | y_n, X_{n-1}^{(i)}\right)$
  - compute $w_n\left(X_{1:n}^{(i)}\right) = \frac{f\left(X_n^{(i)} \middle| X_{n-1}^{(i)}\right) g\left(y_n | X_n^{(i)}\right)}{q\left(X_n^{(i)} \middle| y_n, X_{n-1}^{(i)}\right)}.$
- Resample $\left\{X_{1:n}^{(i)}, W_n^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_{1:n}^{(i)}\right\}$

- Resampling can drastically improve the performance of SIS in models having 'good' mixing properties; e.g. state-space models: this can be verified experimentally *and* theoretically.

- Resampling does not solve all our problems; only the SMC approximations of the most recent marginals $\pi_n\left(x_{n-L+1:n}\right)$ are reliable; i.e. we can have uniform (in time) convergence bounds.

A Limited Framework?

- It seems that there is not much to do to improve over this SMC scheme.
- Given a sequence of distributions $\pi_n(x_{1:n})$, use your favourite resampling scheme and the only degree of freedom is essentially $q_n(x_n | x_{1:n-1})$.
- We know that the 'best' choice is $q_n^{\text{opt}}(x_n | x_{1:n-1}) = \pi_n(x_n | x_{1:n-1})$ so how can we do any better???
- **Answer**: Modify the sequence of target distributions and the associated proposals in a sensible way.

Advanced SMC Methods

- Auxiliary particle filter
- Resample-move algorithm
- Block sampling strategy

Auxiliary Particle Filter

- This is a very popular strategy introduced by Pitt & Shephard (1999).
- It was originally introduced using auxiliary variables but presentation here is completely different...
- **Initial Remark**: The standard SMC algorithm appears very inefficient when

$$q\left(x_n \middle| y_n, x_{n-1}\right) = p\left(x_n \middle| y_n, x_{n-1}\right) \Rightarrow w_n\left(x_{1:n}\right) = p\left(y_n \middle| x_{n-1}\right)$$

Standard Algorithm vs Alternative Strategy

Standard Strategy

- Sample $X_n^{(i)} \sim p\left(\cdot \middle| y_n, X_{n-1}^{(i)}\right)$ and compute $w_n\left(X_{1:n}^{(i)}\right) = p\left(y_n \middle| X_{n-1}^{(i)}\right)$.
- Resample $\left\{X_{1:n}^{(i)}, W_n^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_{1:n}^{(i)}\right\}$.

Alternative Strategy

- Compute $w_n\left(X_{1:n}^{(i)}\right) = p\left(y_n \middle| X_{n-1}^{(i)}\right)$.
- Resample $\left\{X_{1:n-1}^{(i)}, W_n^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_{1:n-1}^{(i)}\right\}$ approx. dist. $p\left(x_{1:n-1} \middle| y_{1:n}\right)$.
- Sample $X_n^{(i)} \sim p\left(\cdot \middle| y_n, X_{n-1}^{(i)}\right)$
- We swap the sampling and resampling steps; this yields more diverse particles at time n; hence intuitively a better estimate.

Auxiliary Particle Filter

- We can only swap the sampling and resampling steps when $q\left(x_n \middle| y_n, x_{n-1}\right) = p\left(x_n \middle| y_n, x_{n-1}\right)$ as the resulting weight is independent of x_n; i.e. we have $w_n\left(x_{1:n}\right) = p\left(y_n \middle| x_{n-1}\right)$.
- If we cannot sample from $p\left(x_n \middle| y_n, x_{n-1}\right)$ and/or do not know $p\left(y_n \middle| x_{n-1}\right)$, we can simply propose to use approximations $\widehat{p}\left(x_n \middle| y_n, x_{n-1}\right)$ and $\widehat{p}\left(y_n \middle| x_{n-1}\right)$ and correct for the bias.
- The APF is thus essentially a look-ahead strategy where we try to anticipate the 'quality' of our current particles $\left\{X_{1:n-1}^{(i)}\right\}$ with respect to y_n.

Algorithm

- Assuming you have $\left\{\overline{W}_{n-1}^{(i)}, X_{1:n-1}^{(i)}\right\}$ approx. $p\left(x_{1:n-1} \middle| y_{1:n-1}\right)$.
- Compute $W_n^{(i)} \propto \overline{W}_{n-1}^{(i)} \widehat{p}\left(y_n \middle| X_{n-1}^{(i)}\right)$.
- Resample $\left\{X_{1:n-1}^{(i)}, W_n^{(i)}\right\}$ to obtain new particles also denoted $\left\{X_{1:n-1}^{(i)}\right\}$ approx. dist.
  $\widetilde{p}\left(x_{1:n-1} \middle| y_{1:n}\right) \propto p\left(x_{1:n-1} \middle| y_{1:n-1}\right) \widehat{p}\left(y_n \middle| x_{n-1}\right)$.
- Sample $X_n^{(i)} \sim \widehat{p}\left(\cdot \middle| y_n, X_{n-1}^{(i)}\right)$ and reweight

$$\begin{aligned}
\overline{W}_n^{(i)} &\propto \frac{p\left(X_{1:n}^{(i)} \middle| y_{1:n}\right)}{p\left(X_{1:n-1}^{(i)} \middle| y_{1:n-1}\right) \widehat{p}\left(y_n \middle| X_{n-1}^{(i)}\right) \widehat{p}\left(X_n^{(i)} \middle| y_n, X_{n-1}^{(i)}\right)} \\
&\propto \frac{f\left(X_n^{(i)} \middle| X_{n-1}^{(i)}\right) g\left(y_n \middle| X_n^{(i)}\right)}{\widehat{p}\left(y_n \middle| X_{n-1}^{(i)}\right) \widehat{p}\left(X_n^{(i)} \middle| y_n, X_{n-1}^{(i)}\right)}
\end{aligned}$$

Interpretation as a standard SMC algorithm

- It is easy to check that this algorithm is nothing but a standard SMC for
$$\pi_n\left(x_{1:n}\right) \propto p\left(x_{1:n}\middle|\, y_{1:n}\right) \widehat{p}\left(y_{n+1}\middle|\, x_n\right)$$
- We do not target $p\left(x_{1:n}\middle|\, y_{1:n}\right)$ directly so it is necessary to use IS to correct for the discrepancy between this target and $\pi_{n-1}\left(x_{1:n-1}\right) \widehat{p}\left(x_n\middle|\, y_n, x_{n-1}\right)$ the distribution of the particles obtained after the sampling step
$$\overline{W}_n^{(i)} \propto \frac{p\left(X_{1:n}^{(i)}\middle|\, y_{1:n}\right)}{p\left(X_{1:n-1}^{(i)}\middle|\, y_{1:n-1}\right) \widehat{p}\left(y_n\middle|\, X_{n-1}^{(i)}\right) \widehat{p}\left(X_n^{(i)}\middle|\, y_n, X_{n-1}^{(i)}\right)}$$
- All the convergence results for standard SMC can thus straightforwardly be extended to the APF.
- Perhaps surprisingly, the APF does not dominate uniformly the standard SMC scheme even if $q\left(x_n\middle|\, y_n, x_{n-1}\right) = p\left(x_n\middle|\, y_n, x_{n-1}\right)$. This is because it is just a one-step optimization procedure.

Design Issues

- In the literature, it is often suggested to approximate

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

  via

$$\widehat{p}\left(y_n \middle| x_{n-1}\right) = g\left(y_n \middle| \mu\left(x_{n-1}\right)\right)$$

  where $\mu\left(x_{n-1}\right)$ is the mode, mean or median of $f\left(x_n \middle| x_{n-1}\right)$.
- Typically, people tend to build an approximation $\widehat{p}\left(x_n \middle| y_n, x_{n-1}\right)$ independently of the approximation $\widehat{p}\left(y_n \middle| x_{n-1}\right)$.
- A simpler and better way consists of building an approximation $\widehat{p}\left(x_n, y_n \middle| x_{n-1}\right) = \widehat{p}\left(x_n, y_n \middle| x_{n-1}\right) \widehat{p}\left(y_n \middle| x_{n-1}\right)$ of $p\left(x_n, y_n \middle| x_{n-1}\right) = g\left(y_n \middle| x_n\right) f\left(x_n \middle| x_{n-1}\right)$ such that

$$\frac{p\left(x_n, y_n \middle| x_{n-1}\right)}{\widehat{p}\left(x_n, y_n \middle| x_{n-1}\right)} < C < \infty$$

Limitations

- The algorithms described earlier suffer from several limitations.
- Even if the optimal importance distribution $p\left(x_n \middle| y_n, x_{n-1}\right)$ can be used, this does not guarantee that the SMC algorithms will be efficient. Indeed, if the variance of $p\left(y_n \middle| x_{n-1}\right)$ is high, then the variance of the resulting approximation will be high. Hence it will be necessary to resample very frequently and the approximation $\widehat{p}\left(x_{1:n} \middle| y_{1:n}\right)$ of the joint distribution $p\left(x_{1:n} \middle| y_{1:n}\right)$ will be unreliable.
- One major problem with the approaches discussed above is that only the variables $\left\{X_n^i\right\}$ are sampled at time n but the path values $\left\{X_{1:n-1}^i\right\}$ remain fixed.
- An obvious way to improve upon these algorithms would involve not only sampling $\left\{X_n^i\right\}$ at time n, but also modifying the values of the paths over a fixed lag $\left\{X_{n-L+1:n-1}^i\right\}$ for $L > 1$ in light of the new observation y_n; L being fixed or upper bounded.

Resample-Move

- The Resample-Move algorithm (Gilks & Berzuini, JRSS B, 2001) is a standard approach to mitigate this problem.
- Like MCMC, it relies upon Markov kernels with appropriate invariant distributions. Whilst MCMC uses such kernels to generate collections of correlated samples, the Resample-Move algorithm uses them in within an SMC algorithm as a principled way to "jitter" the particle locations and thus to reduce degeneracy.
- A Markov kernel $K_n\left(x'_{1:n} \middle| x_{1:n}\right)$ of invariant distribution $p\left(x_{1:n} \middle| y_{1:n}\right)$ is a Markov transition kernel with the property that

$$\int p\left(x_{1:n} \middle| y_{1:n}\right) K_n\left(x'_{1:n} \middle| x_{1:n}\right) dx_{1:n} = p\left(x'_{1:n} \middle| y_{1:n}\right).$$

- For such a kernel, if $X_{1:n} \sim p\left(x_{1:n} \middle| y_{1:n}\right)$ and $X'_{1:n} \middle| X_{1:n} \sim K\left(x_{1:n} \middle| X_{1:n}\right)$ then $X'_{1:n} \sim p\left(x_{1:n} \middle| y_{1:n}\right)$. Even if $X_{1:n}$ is not dist.according to $p\left(x_{1:n} \middle| y_{1:n}\right)$ then, after applying K_n, $X'_{1:n}$ can only have a dist. closer to $p\left(x_{1:n} \middle| y_{1:n}\right)$ in TV than that of $X_{1:n}$.

Examples of Markov Kernels

- For example, we could consider the following Gibbs sampler: set $x'_{1:n-L} = x_{1:n-L}$ then sample x'_{n-L+1} from $p\left(x_{n-L+1} \middle| y_{1:n}, x'_{1:n-L}, x_{n-L+2:n}\right)$, sample x'_{n-L+2} from $p\left(x_{n-L+2} \middle| y_{1:n}, x'_{1:n-L+1}, x_{n-L+3:n}\right)$ and so on until we sample x'_n from $p\left(x_n \middle| y_{1:n}, x'_{1:n-1}\right)$; that is

$$K_n\left(x'_{1:n} \middle| x_{1:n}\right) = \delta_{x_{1:n-L}}\left(x'_{1:n-L}\right) \prod_{k=n-L+1}^{n} p\left(x'_k \middle| y_{1:n}, x'_{1:k-1}, x_{k+1:n}\right)$$

and we write, with a slight abuse of notation, the non-degenerate component of the MCMC kernel $K_n\left(x'_{n-L+1:n} \middle| x_{1:n}\right)$. It is straightforward to verify that this kernel is $p\left(x_{1:n} \middle| y_{1:n}\right)$-invariant.

- If it is not possible to sample from $p\left(x'_k \middle| y_{1:n}, x'_{1:k-1}, x_{k+1:n}\right) = p\left(x'_k \middle| y_k, x'_{k-1}, x_{k+1}\right)$, we can instead employ a Metropolis-Hastings (MH) strategy and sample a candidate according to some proposal $q\left(x'_k \middle| y_k, x'_{k-1}, x_{k:k+1}\right)$ and accept it with the usual MH acceptance probability

$$\min\left(1, \frac{p\left(x'_{1:k}, x_{k+1:n} \middle| y_{1:n}\right) q\left(x_k \middle| y_k, x'_{k-1}, x'_k, x_{k+1}\right)}{p\left(x'_{1:k-1}, x_{k+1:n} \middle| y_{1:n}\right) q\left(x'_k \middle| y_k, x'_{k-1}, x_{k:k+1}\right)}\right)$$

$$= \min\left(1, \frac{g\left(y_k \middle| x'_k\right) f\left(x_{k+1} \middle| x'_k\right) f\left(x'_k \middle| x'_{k-1}\right) q\left(x_k \middle| y_k, x'_{k-1}, x'_k, x_{k+1}\right)}{g\left(y_k \middle| x_k\right) f\left(x_{k+1} \middle| x_k\right) f\left(x_k \middle| x'_{k-1}\right) q\left(x'_k \middle| y_k, x'_{k-1}, x_{k:k+1}\right)}\right)$$

- These kernels can be ergodic only if $L = n$ so that *all* of the components of $x_{1:n}$ are updated. However, in our context we will use non-ergodic kernels as we restrict ourselves to updating the variables $X_{n-L+1:n}$ for some fixed or bounded L.

Resample Move

Assuming we have access to $\left\{X'^i_{1:n-1}\right\}$ approx. dist. $p\left(x_{1:n-1}\middle|\, y_{1:n-1}\right)$ then at times $n \geq L$

- Sample $X^i_n \sim q\left(x_n\middle|\, y_n, X'^i_{n-1}\right)$ and set $X^i_{1:n} \leftarrow \left(X'^i_{1:n-1}, X^i_n\right)$.
- Compute the weights $W^i_n \propto \frac{g\left(y_n|X^i_n\right) f\left(X^i_n\middle|X^i_{n-1}\right)}{q\left(X^i_n|y_n,X^i_{n-1}\right)}$.
- Resample $\left\{W^i_n, X^i_{1:n}\right\}$ to obtain N new equally-weighted particles $\left\{\frac{1}{N}, \overline{X}^i_{1:n}\right\}$.
- Sample $X'^i_{n-L+1:n} \sim K_n\left(x_{n-L+1:n}\middle|\, \overline{X}^i_{1:n}\right)$ and set $X'^i_{1:n} \leftarrow \left(\overline{X}^i_{1:n-L}, X'^i_{n-L+1:n}\right)$.

Interpretation as a standard SMC algorithm

- We can justify inserting MCMC transitions within an SMC algorithm as follows. Given a target distribution π, an instrumental distribution μ and a π-invariant Markov kernel K, the following generalization of the IS identity is trivially true:

$$\begin{aligned} & \int \pi(y)\varphi(y)dy \\ = & \iint \mu(x)K(y|x)\frac{\pi(y)L(x|y)}{\mu(x)K(y|x)}\varphi(y)dxdy \end{aligned}$$

  for any Markov kernel L.

- This approach corresponds to IS on an enlarged space using $\mu(x)K(y|x)$ as the proposal distribution for a target $\pi(y)L(x|y)$ and then estimating a function $\varphi'(x,y) = \varphi(y)$.

- In particular, for the time-reversal kernel associated with K

$$L(x|y) = \frac{\pi(x)K(y|x)}{\pi(y)},$$

we have the importance weight

$$\frac{\pi(y)L(x|y)}{\mu(x)K(y|x)} = \frac{\pi(x)}{\mu(x)}.$$

- This interpretation of such an approach illustrates its deficiency: the importance weights depend only upon the location before the MCMC move while the sample depends upon the location after the move.
- Even if the kernel was perfectly mixing, leading to a collection of iid samples from the target distribution, some of these samples would be eliminated and some replicated in the resampling step.
- Resampling after an MCMC step will always lead to greater sample diversity than performing the steps in the other order (and this algorithm can be justified directly by the invariance property).

- It is possible to reformulate this algorithm as a specific application of the generic SMC algorithm.
- To simplify notation we write $q_n\left(x_n \middle| x_{n-1}\right)$ for $q\left(x_n \middle| y_n, x_{n-1}\right)$ and to clarify our argument, it is necessary to add a superscript to the variables; e.g. X_k^p corresponds to the p^{th} time the random variable X_k is sampled; in this and the following section, this superscript *does not* denote the particle index.
- Resample move is a generic SMC algorithm associated to

$$\begin{aligned}
&\pi_n\left(x_1^{1:L+1}, \ldots, x_{n-L+1}^{1:L+1}, x_{n-L}^{1:L}, \ldots, x_n^{1:2}\right) \\
&= p\left(x_1^{L+1}, \ldots., x_{n-L+1}^{L+1}, x_{n-L}^{L}, \ldots, x_n^{2} \middle| y_{1:n}\right) \\
&\times L_n\left(x_n^1, x_{n-1}^2, \ldots, x_{n-L+1}^{L} \middle| x_n^2, x_{n-1}^3, \ldots, x_{n-L+1}^{L+1}\right) \\
&\times \cdots \times K_2\left(x_1^2, x_2^1 \middle| x_1^3, x_2^2\right) L_1\left(x_1^1 \middle| x_1^2\right)
\end{aligned}$$

where L_n is the time-reversal kernel associated with K_n.

- If no resampling is used then we have

$$
\begin{aligned}
& q_n\left(x_1^{1:L+1}, \ldots, x_{n-L+1}^{1:L+1}, x_{n-L}^{1:L}, \ldots, x_n^{1:2}\right) \\
& = q_1\left(x_1^1\right) K_1\left(x_1^2 \middle| x_1^1\right) q_2\left(x_2^1 \middle| x_1^2\right) K_2\left(x_1^3, x_2^2 \middle| x_1^2, x_2^1\right) \\
& \times \cdots \times q_n\left(x_n^1 \middle| x_{n-1}^2\right) \\
& \times K_n\left(x_{n-L+1}^{L+1}, \ldots, x_{n-1}^3, x_n^2 \middle| x_{1:n-L}^{L+1}, x_{n-L+1}^L, \ldots, x_{n-1}^2, x_n^1\right).
\end{aligned}
$$

- This sequence of target distributions admits the filtering distributions of interest as marginals.
- The clear theoretical advantage of using MCMC moves is that the use of even non-ergodic MCMC kernels $\{K_n\}$ can only improve the mixing properties of $\{\pi_n\}$ compared to the "natural" sequence of filtering distributions; this explains why these algorithms outperform a standard particle filter for a given number of particles.

Block Sampling

- Resample-Move suffers from a major drawback: it does allows us to reintroduce some diversity among the set of particles after the resampling step over a lag of length $L > 1$, the importance weights have the same expression as for the standard particle filter.
- This strategy does not significantly decrease the number of resampling steps compared to a standard approach. It can partially mitigate the problem associated with resampling, but it does not prevent these resampling steps in the first place.
- An alternative approach block sampling approach consists of directly sampling the components $x_{n-L+1:n}$ at time n; the previously-sampled values of the components $x_{n-L+1:n-1}$ sampled are simply discarded.

Why it is not trivial...

- The basic idea is trivial but not applicable....
- Consider you have $X_{1:n-1} \sim p\left(x_{1:n-1} \middle| y_{1:n-1}\right)$ and at time n you sample $X'_{n-L+1:n} \sim q\left(\cdot \middle| X_{n-L}, y_{n-L+1:n}\right)$ then the joint distribution of $\left(X_{1:n-1}, X'_{n-L+1:n}\right)$

$$p\left(x_{1:n-1} \middle| y_{1:n-1}\right) q\left(x'_{n-L+1:n} \middle| x_{n-L}, y_{n-L+1:n}\right)$$

- If we discard $X_{n-L+1:n-1}$, then the distribution of $\left(X_{1:n-L}, X'_{n-L+1:n}\right)$ is

$$\begin{aligned} & \int p\left(x_{1:n-1} \middle| y_{1:n-1}\right) q\left(x'_{n-L+1:n} \middle| x_{n-L}, y_{n-L+1:n}\right) dx_{n-L+1:n-1} \\ = \quad & p\left(x_{1:n-L} \middle| y_{1:n-1}\right) q\left(x'_{n-L+1:n} \middle| x_{n-L}, y_{n-L+1:n}\right). \end{aligned}$$

- We typically do not know $p\left(x_{1:n-L} \middle| y_{1:n-1}\right)$ up to a normalizing constant so we cannot use IS!

Extended Importance Sampling

- The idea consists of using an extended target distribution

$$p\left(x_{1:n-L}, x'_{n-L+1:n} \middle| y_{1:n}\right) \overline{q}(x_{n-L+1:n-1} | y_{n-L+1:n-1}, x_{n-L})$$

  whose marginal is by construction $p\left(x_{1:n-L}, x'_{n-L+1:n} \middle| y_{1:n}\right)$.
- We then use IS on this extended state-space and compute the weights

$$\frac{p\left(x_{1:n-L}, x'_{n-L+1:n} \middle| y_{1:n}\right) \overline{q}(x_{n-L+1:n-1} | y_{n-L+1:n-1}, x_{n-L})}{p\left(x_{1:n-1} \middle| y_{1:n-1}\right) q\left(x'_{n-L+1:n} \middle| x_{n-L}, y_{n-L+1:n}\right)}$$

Algorithm Settings

- If we knew how to compute $p\left(x_{1:n-L} \middle| y_{1:n-1}\right)$ then the IS distribution minimizing the variance of the importance weight is

$$q^{\text{opt}}\left(x_{n-L+1:n} \middle| x_{n-L}, y_{n-L+1:n}\right) = p\left(x_{n-L+1:n} \middle| y_{n-L+1:n}, x_{n-L}\right)$$

  and the importance weight is

$$p\left(y_{n-L+1:n} \middle| x_{n-L}\right) = \int \prod_{k=n-L+1}^{n} f\left(x_k \middle| x_{k-1}\right) \cdot g\left(y_k \middle| x_k\right) dx_{n-L+1:n}.$$

- This optimal weight has a variance which typically decreases exponentially fast with L (under mixing assumptions).
- As this distribution is typically not available and/or $p\left(y_{n-L+1:n} \middle| x_{n-L}\right)$ cannot be computed, we need to use an approximation.

- The distribution $\overline{q}(x_{n-L+1:n-1}|y_{n-L+1:n-1}, x_{n-L})$ minimizing the variance of the weights is simply

$$\begin{aligned}
&\overline{q}^{\text{opt}}(x_{n-L+1:n-1}|y_{n-L+1:n-1}, x_{n-L}) \\
= \quad & \frac{p\left(x_{1:n-1}\middle|y_{1:n-1}\right) q\left(x'_{n-L+1:n}\middle|x_{n-L}, y_{n-L+1:n}\right)}{\int p\left(x_{1:n-1}\middle|y_{1:n-1}\right) q\left(x'_{n-L+1:n}\middle|x_{n-L}, y_{n-L+1:n}\right) dx_{n-L+1:n-1}}
\end{aligned}$$

- So if we pick
  $q^{\text{opt}}\left(x_{n-L+1:n}\middle|x_{n-L}, y_{n-L+1:n}\right) = p\left(x_{n-L+1:n}\middle|y_{n-L+1:n}, x_{n-L}\right)$ then

  $\overline{q}^{\text{opt}}(x_{n-L+1:n-1}|y_{n-L+1:n-1}, x_{n-L}) = p(x_{n-L+1:n-1}|y_{n-L+1:n-1}, x_{n-L})$

- This suggests once more using an approximation of this density

Block Sampling SMC

Assuming we have access to $\left\{\overline{X}^i_{1:n-1}\right\}$ approx. dist. $p\left(x_{1:n-1}\middle|\, y_{1:n-1}\right)$ then at time $n \geq L$

- Sample $X^i_{n-L+1:n} \sim q(x_{n-L+1:n}|y_{n-L+1:n}, \overline{X}^i_{1:n-1})$.
- Compute the weights

$$W^i_n \propto \frac{p\left(\overline{X}^i_{1:n-L}, X^i_{n-L+1:n}, y_{1:n}\right)\widehat{p}(\overline{X}^i_{n-L+1:n-1}|y_{n-L+1:n-1}, \overline{X}^i_{n-L})}{p\left(\overline{X}^i_{1:n-1}, y_{1:n-1}\right)\widehat{p}(X^i_{n-L+1:n}|y_{n-L+1:n}, \overline{X}^i_{n-L})}.$$

- Resample $\left\{W^i_n, \overline{X}^i_{1:n-L}, X^i_{n-L+1:n}\right\}$ to obtain N new equally weighted particles $\left\{\overline{X}^i_{1:n}\right\}$.

Interpretation as a standard SMC algorithm

- Once more this is just a special case of the generic SMC algorithm.
- To simply notation we write $q_n\left(x_{n-L+1:n}\middle|\, x_{n-L}\right)$ for $q\left(x_{n-L+1:n}\middle|\, y_{n-L+1:n}, x_{n-L}\right)$ and to clarify our argument we add a superscript to the variables; e.g. X_k^p corresponds to the p^{th} time the random variable X_k is sampled.
- The block sampling algoritmh corresponds to

$$\begin{aligned}&\pi_n\left(x_1^{1:L}, \ldots, x_{n-L+1}^{1:L}, x_{n-L+2}^{1:L-1}, \ldots, x_n^1\right)\\&= p\left(x_{1:n-L+1}^L, x_{n-L+2}^{L-1}, \ldots, x_n^1\middle|\, y_{1:n}\right) q_{n-1}\left(x_{n-L+1}^{L-1}, \ldots, x_{n-1}^1\middle|\, x_{n-L}^L\right)\\&\times \cdots q_2\left(x_1^2, x_2^1\right) q_1\left(x_1^1\right).\end{aligned}$$

- If no resampling is used, a path is sampled according to

$$\begin{aligned}&q_n\left(x_1^{1:L}, \ldots, x_{n-L+1}^{1:L}, x_{n-L+2}^{1:L-1}, \ldots, x_n^1\right)\\&= q_1\left(x_1^1\right) q_2\left(x_1^2, x_2^1\right) \times \cdots \times q_n\left(x_{n-L+1}^L, \ldots, x_n^1\middle|\, x_{n-L}^L\right).\end{aligned}$$

Application to Bearing-only Tracking

- Target model

$$X_n = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix} X_{n-1} + V_n$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma)$.

- The state vector $X_n = \begin{pmatrix} X_n^1 & X_n^2 & X_n^3 & X_n^4 \end{pmatrix}^{\mathsf{T}}$ is such that X_n^1 (resp. X_n^3) corresponds to the horizontal (resp. vertical) position of the target whereas X_n^2 (resp. X_n^4) corresponds to the horizontal (resp. vertical) velocity. One only receives observations of the bearings of the target from a sensor located at the origin

$$Y_n = \tan^{-1}\left(\frac{X_n^3}{X_n^1}\right) + W_n$$

where $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, 10^{-4}\right)$; i.e. the observations are almost noiseless.

- We build an approximation $\widehat{p}\left(x_{n-L+1:n} \middle| y_{n-L+1:n}, x_{n-L}\right)$ of the optimal importance distribution using the EKF and the forward filtering/backward sampling formula.
- We compare the standard bootstrap filter, two resample-move algorithms where the SISR algorithm for $L = 1$ using the EKF proposal is used followed by: (i) one at a time Metropolis-Hastings (MH) moves using an approximation of the full conditionals $p\left(x_k \middle| y_k, x_{k-1}, x_{k+1}\right)$ as a proposal over a lag $L = 10$ (algorithm RML(10)); and (ii) using the EKF proposal for $L = 10$ (algorithm RMFL(10)), the block sampling algorithms for $L = 1, 2, 5$ and 10 which are using the EKF proposal denoted SMC-EKF(L).
- Systematic resampling is performed whenever the ESS goes below $N/2$.

Comparison in terms of resampling steps

Filter	Avge. # Resampling steps
Bootstrap	46.7
SMC-EKF(1)	44.6
RML(10)	45.2
RMFL(10)	43.3
SMC-EKF(2)	34.9
SMC-EKF(5)	4.6
SMC-EKF(10)	1.3

Table 1: Average number of resampling steps for 100 simulations, 100 time instances per simulation using $N = 1000$ particles.

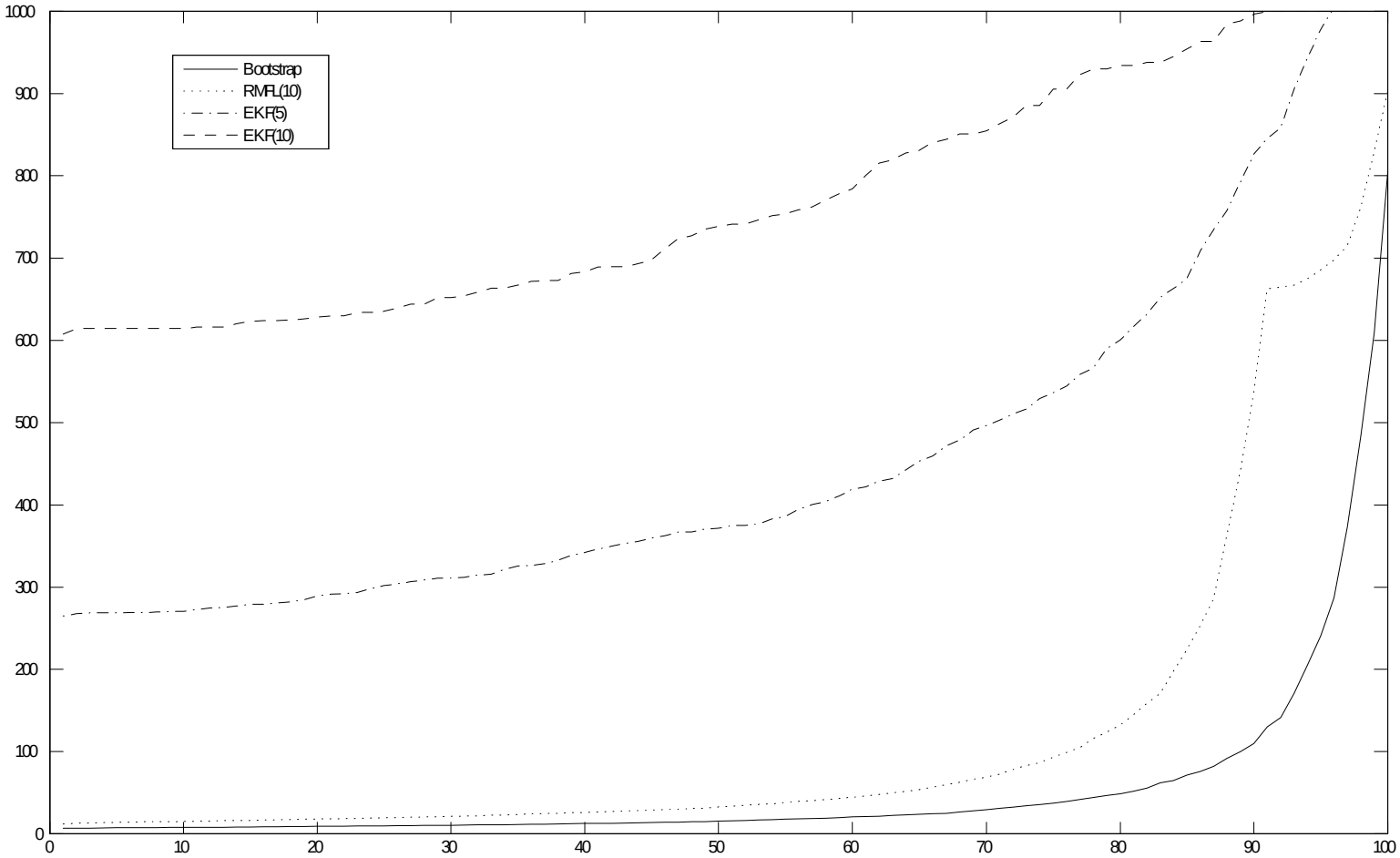

Figure: Average number of unique particles $X_n^{(i)}$ approximating $p\left(x_n \middle| y_{1:100}\right)$ plotted against time (x-axis)

Application to Stochastic Volatility

- We consider a standard SV model

$$X_n = \phi X_{n-1} + \sigma V_n, \; X_1 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right),$$

$$Y_n = \beta \exp\left(X_n/2\right) W_n,$$

  where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$.
- We propose to build approximation of $p\left(x_{n-L+1:n} \middle| y_{n-L+1:n}, x_{n-L}\right)$ using the fact that

$$\log\left(Y_n^2\right) = \log\left(\beta^2\right) + X_n + \log\left(W_n^2\right).$$

- We approximate non-Gaussian noise term $\log\left(W_n^2\right)$ with a Gaussian noise of similar mean and variance and hence obtain a linear Gaussian model approximation. We then use the KF to build our proposal.
- The performance of our algorithms are assessed through computer simulations based on varying samples sizes to attain an approximately *equal* computational cost.

Filter	# Particles	Avge. # Resampling Steps
Bootstrap	50000	176.2
SMC-EKF(1)	12000	127.1
SMC-EKF(2)	4000	80.0
SMC-EKF(5)	1600	11.6
SMC-EKF(10)	1000	0.45

Table 2: Average number of resampling steps for 100 simulations using 500 time instances per simulation.

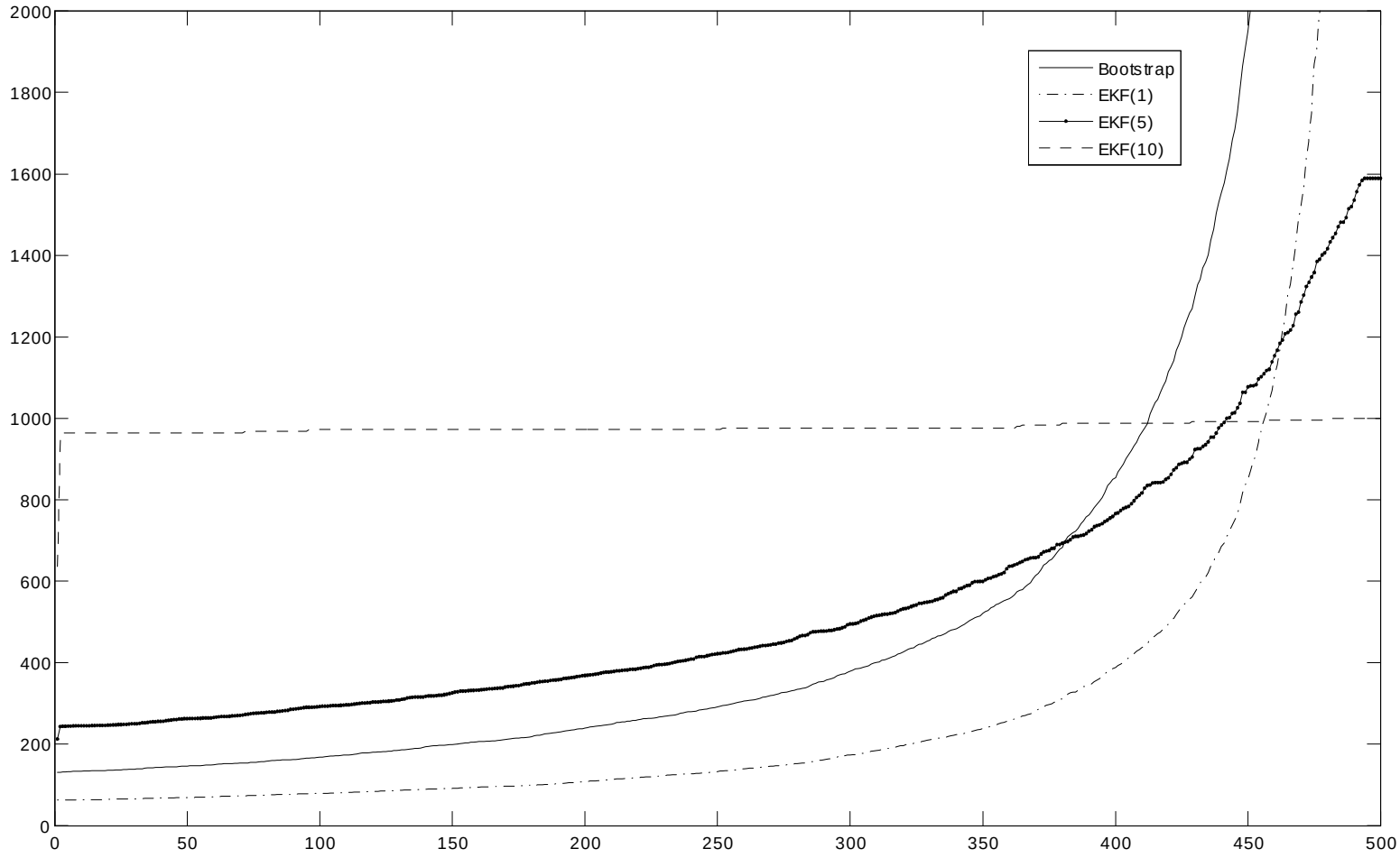

Figure: Average number of unique particles $X_n^{(i)}$ approximating $p\left(x_n \middle| y_{1:100}\right)$ plotted against time (x-axis)

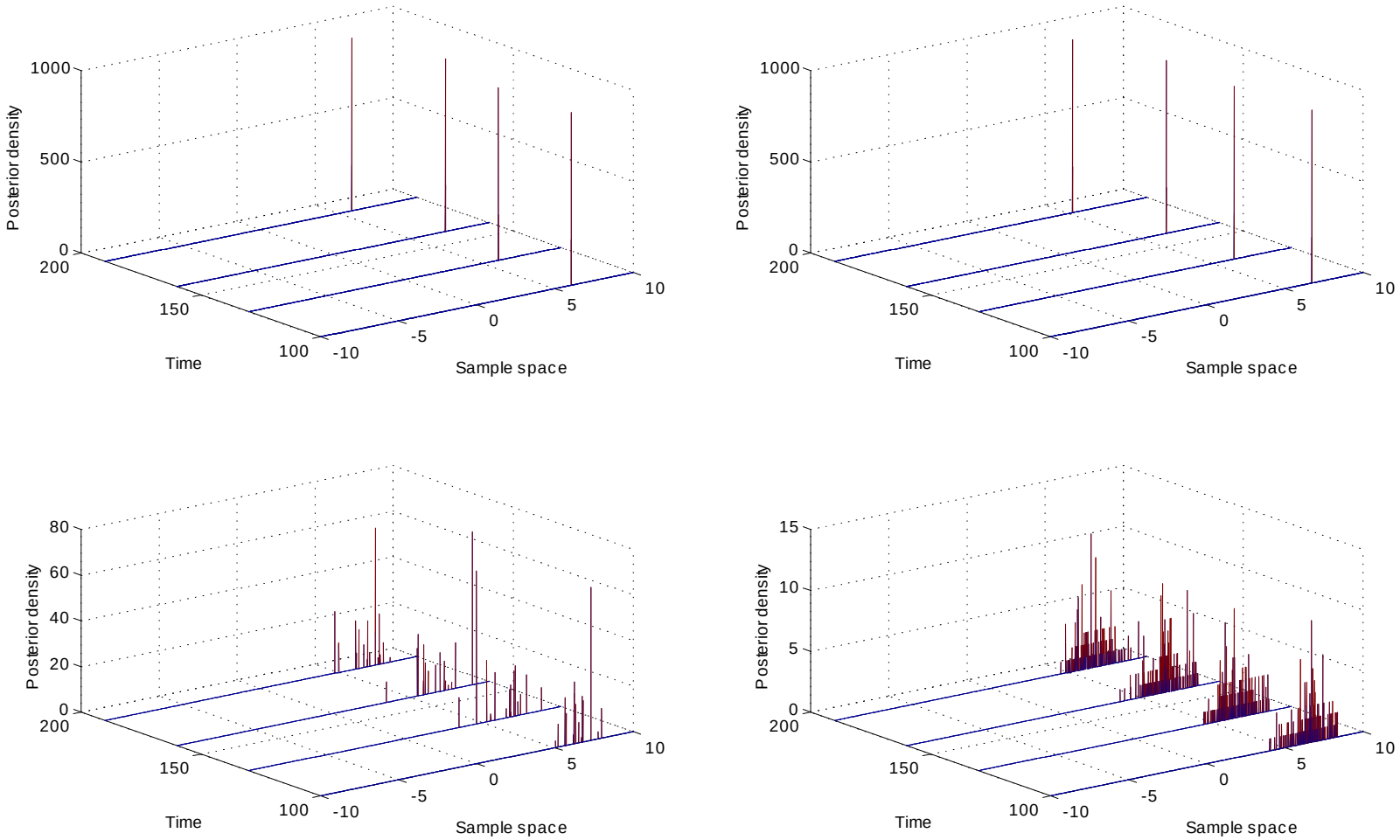

Figure: Empirical measure approximations of $p\left(x_n \middle| y_{1:945}\right)$ at times $n = 100, 130, 160, 190$ for Bootstrap (top left), SMC-EKF(1) (top right), SMC-EKF(5) (bottom left), SMC-EKF(10) (bottom right)