Divide-and-Conquer sequential Monte Carlo with applications to high dimensional filtering

Francesca R. Crucinio

11 May 2023

CSML Reading Group, Lancaster

Outline

1. Sequential Monte Carlo (SMC)
2. Divide-and-Conquer SMC (DaC-SMC)
3. High Dimensional Filtering

Sequential Monte Carlo (SMC)

Sequential Monte Carlo (SMC) methods approximate sequences of targets $\{\pi_t\}_{t\geq 1}$ of the form

$$\pi_t(x_{1:t}) \propto \pi_{t-1}(x_{1:t-1}) M_t(x_{t-1}, x_t) G_t(x_t),$$

where M_t are Markov kernels and G_t are non-negative weight functions (Chopin and Papaspiliopoulos, 2020).

Can be used

- as an alternative to MCMC (Del Moral et al., 2006)
- for rare event simulation (Cérou et al., 2012)
- for ABC (Sisson et al., 2007)
- for filtering

State Space Models (SSM)

$$
\begin{tikzcd}
 & Y_{t-1} & Y_t & \\
\cdots \arrow[r] & X_{t-1} \arrow[u] \arrow[r, "f_t"] & X_t \arrow[u, "g_t"'] \arrow[r] & \cdots
\end{tikzcd}
$$

A state space model $(X_t, Y_t)_{t \geq 1} \subset \mathbb{R}^{d \times p}$ with

- transition density for the *latent* process $f_t(x_{t-1}, x_t)$
- likelihood for the *observation* process $g_t(x_t, y_t)$

Consider

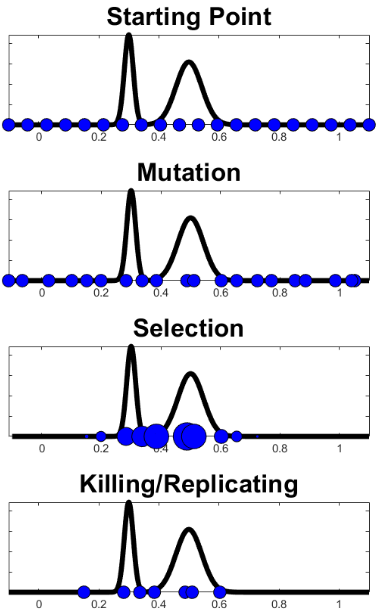
$$
\begin{aligned}
p(x_{1:t} | y_{1:t}) &\propto \prod_{k=1}^{t} f_k(x_{k-1}, x_k) g_k(x_k, y_k) \\
&= p(x_{1:t-1}, y_{1:t-1}) f_t(x_{t-1}, x_t) g_t(x_t, y_t).
\end{aligned}
$$

Sequential Monte Carlo (SMC)

- **Aim:** Approximate $\pi_t(x_t)$ using a set of N weighted particles $\{X_t^i, W_t^i\}_{i=1}^N$
- **Ingredients:**
    1. a Markov kernel to sample $X_t^i \sim M_t(\tilde{X}_{t-1}^i, \cdot)$
    2. a positive weight function to compute the weights $W_t^i \propto G_t(X_t^i)$
    3. a resampling scheme to obtain $\{\tilde{X}_t^i, 1/N\}_{i=1}^N$ from $\{X_t^i, W_t^i\}_{i=1}^N$

Example: the bootstrap particle filter corresponds to $\pi_t(x_t) = p(x_t|y_{1:t})$, $M_t = f_t$ and $G_t = g(y_t|\cdot)$.

Sequential Monte Carlo (SMC)



Outline

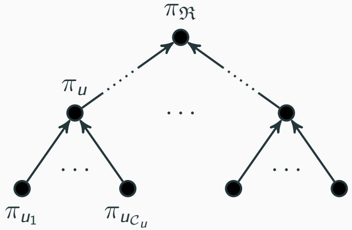
1 Sequential Monte Carlo (SMC)

2 Divide-and-Conquer SMC (DaC-SMC)

3 High Dimensional Filtering

Divide-and-Conquer SMC (DaC-SMC)

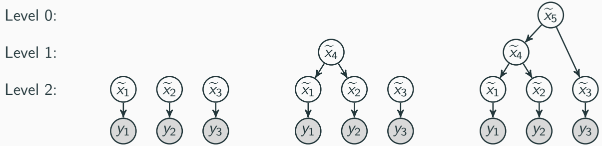
An extension of standard SMC in which the sequence of target distributions $\{\pi_u\}_{u \in \mathbb{T}}$ evolves on a tree $\mathbb{T}$ rather than on a line (Lindsten et al., 2017).



For simplicity we will also consider the unnormalised targets $\{\gamma_u\}_{u \in \mathbb{T}}$, s.t. $\pi_u = \gamma_u / Z_u$.

Divide-and-Conquer SMC (DaC-SMC)

- easier to distribute/parallelise (Corneflos et al., 2022; Ding and Gandy, 2018)
- some models more naturally adapted to a tree structure than a ‘linear” one: e.g. graphical models (Lindsten et al., 2017; Paige and Wood, 2016; Jewell, 2015)



- distributed inference (Chan et al., 2021; Manderson and Goudie, 2021)

Divide-and-Conquer SMC (DaC-SMC)

At each node u we have a set of N weighted particles $\{X_u^i, W_u^i\}_{i=1}^N$ approximating π_u.

To evolve use standard SMC ingredients

- transition/mutation: $X_u^i \sim M_u((X_{\ell(u)}^i, X_{r(u)}), \cdot)$
- reweighting: $W_u^i \propto G_u(X_u^i)$
- resampling

with the addition of a **merging** step when two branches of the tree merge.

The Merge Step

- **Aim:** Given two populations of weighted particles on the left and the right child, $\{X^i_{\ell(u)}, W^i_{\ell(u)}\}_{i=1}^N$ and $\{X^i_{r(u)}, W^i_{r(u)}\}_{i=1}^N$, approximating $\gamma_{\ell(u)}$ and $\gamma_{r(u)}$ build an approximation of γ_u

The Merge Step

- **Aim:** Given two populations of weighted particles on the left and the right child, $\{X^i_{\ell(u)}, W^i_{\ell(u)}\}_{i=1}^N$ and $\{X^i_{r(u)}, W^i_{r(u)}\}_{i=1}^N$, approximating $\gamma_{\ell(u)}$ and $\gamma_{r(u)}$ build an approximation of γ_u
- **Ingredients:**
  - the (weighted) product form estimator (Kuntz et al., 2022)

$$\gamma^N_{\ell(u)} \times \gamma^N_{r(u)} = \frac{1}{N^2} \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} W^{i_1}_{\ell(u)} W^{i_2}_{r(u)} \delta_{(X^{i_1}_{\ell(u)}, X^{i_2}_{r(u)})}$$

  - N^2 *mixture* (importance) weights

$$m_u^{(i_1,i_2)} := W^{i_1}_{\ell(u)} W^{i_2}_{r(u)} \frac{\gamma_u(x^{i_1}_{\ell(u)}, x^{i_2}_{r(u)})}{\gamma_{\ell(u)}(x^{i_1}_{\ell(u)})\gamma_{r(u)}(x^{i_2}_{r(u)})}$$

  - a resampling scheme to obtain $\{\tilde{X}^i_u, 1/N\}_{i=1}^N$ from $\{(X^{i_1}_{\ell(u)}, X^{i_2}_{r(u)}), m_u^{(i_2,i_2)}\}_{i_1,i_2=1}^N$

Cheaper Alternatives

- lightweight mixture resampling (Lindsten et al., 2017)
- lazy resampling schemes (Corneflos et al., 2022)
- strategies borrowed from the literature on incomplete U-statistics (Kuntz et al., 2021)

Lightweight Mixture Resampling (Lindsten et al., 2017)

- Fix $\theta \ll N$.
- Sample $\{X_{\ell(u)}^{i_1}\}_{i_1=1}^{\theta}$ from $\gamma_{\ell(u)}$ and $\{X_{r(u)}^{i_2}\}_{i_2=1}^{\theta}$ from $\gamma_{r(u)}$.
- Build the θ pairs $(X_{\ell(u)}^{i_1}, X_{r(u)}^{i_2})$ and compute their weights $m_u^{(i_1,i_2)}$.

Adaptive Lightweight Mixture Resampling

A strategy based on the effective sample size $\left(\sum_n m_u^n\right)^2 / \sum_n (m_u^n)^2$.

- Set a target ESS, $\mathrm{ESS}^\star$.
- For $i = 1, \ldots, N$ build $(X^i_{\ell(u)}, X^i_{r(u)})$ and compute their weights $m_u^{(i,i)}$.
- While $\mathrm{ESS} < \mathrm{ESS}^\star$:
  - Draw one permutation of N, $\pi(N)$, build $(X^i_{\ell(u)}, X^{\pi(i)}_{r(u)})$ and compute their weights $m_u^{(i,\pi(i))}$.
  - Update ESS.

Adaptive Lightweight Mixture Resampling

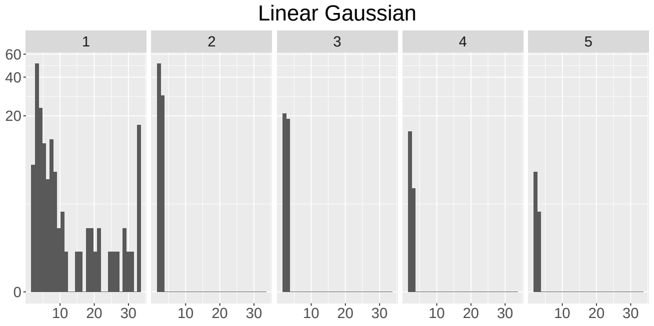


Figure 1: Distribution of the number of permutations selected for a simple linear Gaussian model with $d = 32$, $N = 10^3$, and 10 time steps; the panels correspond to the levels of the tree from the level above the leaves (level 1) to the root (level 5).

Theoretical Properties (Kuntz et al., 2021)

- strong law of large numbers

$$\frac{1}{N}\sum_{i=1}^{N}\varphi(\tilde{X}_u^i) \stackrel{a.s.}{\to} \int \varphi(x)\pi_u(x)\mathrm{d}x$$

- central limit theorem

$$\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^{N}\varphi(\tilde{X}_u^i) - \int \varphi(x)\pi_u(x)\mathrm{d}x\right) \stackrel{d}{\to} \mathcal{N}(0, \sigma_u^2(\varphi))$$

- unbiasedness of normalising constant estimates

And other: $\mathbb{L}^p$ errors, bias estimates, optimal intermediate targets, optimal proposals, . . .

Outline

1 Sequential Monte Carlo (SMC)

2 Divide-and-Conquer SMC (DaC-SMC)

3 High Dimensional Filtering

- Main Ideas
- Experiments

State Space Models (SSM)

$$
\begin{tikzcd}
 & Y_{t-1} & Y_t & \\
\cdots \arrow[r] & X_{t-1} \arrow[u] \arrow[r, "f_t"] & X_t \arrow[u, "g_t"'] \arrow[r] & \cdots
\end{tikzcd}
$$

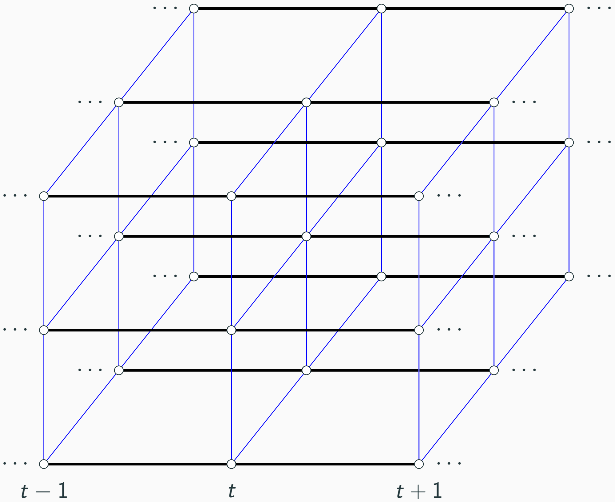
A state space model $(X_t, Y_t)_{t \geq 1} \subset \mathbb{R}^{d \times p}$ with

- transition density for the *latent* process $f_t(x_{t-1}, x_t)$
- likelihood for the *observation* process $g_t(x_t, y_t)$

We are interested in

$$
\begin{aligned}
p(x_{1:t} | y_{1:t}) &\propto \prod_{k=1}^{t} f_k(x_{k-1}, x_k) g_k(x_k, y_k) \\
&= p(x_{1:t-1}, y_{1:t-1}) f_t(x_{t-1}, x_t) g_t(x_t, y_t).
\end{aligned}
$$

Spatial SSM



1 Sequential Monte Carlo (SMC)

2 Divide-and-Conquer SMC (DaC-SMC)

3 High Dimensional Filtering

- Main Ideas
- Experiments

Divide and Conquer SMC for Filtering

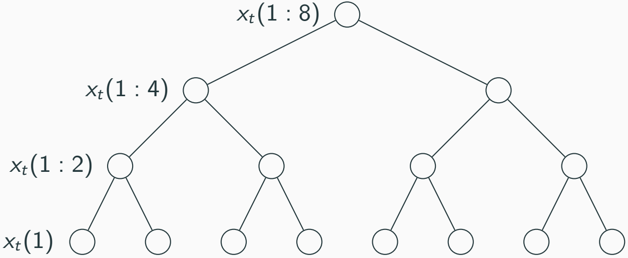


Figure 2: Space decomposition for $d = 8$.

Choice of Distributions

Fix t. We need $\{\gamma_{t,u}\}_{u \in \mathbb{T}}$.

Constraints:

- $\gamma_{t,\mathfrak{R}}(x_t) \propto p(x_t|y_{1:t})$
- for $u \neq \mathfrak{R}$, $\gamma_{t,u}$ only depends on a subset of $x_t(1:d)$
- we can compute

$$\frac{\gamma_{t,u}(x_{t,\ell(u)}, x_{t,r(u)})}{\gamma_{t,\ell(u)}(x_{t,\ell(u)})\gamma_{t,r(u)}(x_{t,r(u)})}$$

Choice of Distributions

Recall

$$p(x_t|y_{1:t}) \propto g_t(x_t, y_t) \int f_t(x_{t-1}, x_t) p(x_{t-1}|y_{1:t-1}) \mathrm{d}x_{t-1}.$$

Use auxiliary functions $f_{t,u}, g_{t,u}$ such that $f_{t,\mathfrak{R}} = f_t, g_{t,\mathfrak{R}} = g_t$ and for $u \in \mathbb{T} \setminus \mathfrak{R}$, $f_{t,u}$ and $g_{t,u}$ serve as proxies for marginals of f_t, g_t

$$\gamma_{t,u}(x_{t,u}) = g_{t,u}(x_{t,u}, (y_t(i))_{i \in u}) \int f_{t,u}(x_{t-1}, x_{t,u}) \gamma_{t-1,\mathfrak{R}}(x_{t-1}) \mathrm{d}x_{t-1}$$

Choice of Distributions

Recall

$$p(x_t|y_{1:t}) \propto g_t(x_t, y_t) \int f_t(x_{t-1}, x_t) p(x_{t-1}|y_{1:t-1}) \mathrm{d}x_{t-1}.$$

Use auxiliary functions $f_{t,u}, g_{t,u}$ such that $f_{t,\mathfrak{R}} = f_t, g_{t,\mathfrak{R}} = g_t$ and for $u \in \mathbb{T} \setminus \mathfrak{R}$, $f_{t,u}$ and $g_{t,u}$ serve as proxies for marginals of f_t, g_t

$$\gamma_{t,u}(x_{t,u}) = g_{t,u}(x_{t,u}, (y_t(i))_{i \in u}) \int f_{t,u}(x_{t-1}, x_{t,u}) \gamma_{t-1,\mathfrak{R}}(x_{t-1}) \mathrm{d}x_{t-1}$$

The integral above is intractable, so we define approximate targets as in marginal particle filters (Klaas et al., 2005)

$$\gamma_{t,u}(x_{t,u}) = g_{t,u}(x_{t,u}, (y_t(i))_{i \in u}) W_{t-1}^i \sum_{i=1}^{N} f_{t,u}(X_{t-1,\mathfrak{R}}^i, x_{t,u})$$

Divide and Conquer SMC for Filtering

At time t:

- start with $\{X^i_{t-1,\mathfrak{R}}, W^i_{t-1}\}_{i=1}^N$ approximating $p(x_{t-1}|y_{1:t-1})$ at the leaf level
- define targets

$$\gamma_{t,u}(x_{t,u}) = g_{t,u}(x_{t,u}, (y_t(i))_{i\in u}) W^i_{t-1} \sum_{i=1}^N f_{t,u}(X^i_{t-1,\mathfrak{R}}, x_{t,u})$$

- use DaC to move up the space from the leaves to the root, i.e. from d particle populations approximating 1-dimensional to one d-dimensional population $\{X^i_{t,\mathfrak{R}}, W^i_t\}_{i=1}^N$ which approximates $p(x_t|y_{1:t})$

Pros and Cons

Pros

- No need for analytical form of $p(x_t(i) \mid x_t(1 : i - 1))$
- No need for factorised likelihoods
- Easy to parallelise and distribute

Cons

- Polynomial cost in N (can be mitigated via GPUs)
- Needs specification of $\gamma_{t,u}$

1 Sequential Monte Carlo (SMC)

2 Divide-and-Conquer SMC (DaC-SMC)

3 High Dimensional Filtering

- Main Ideas
- Experiments

Linear Gaussian Model

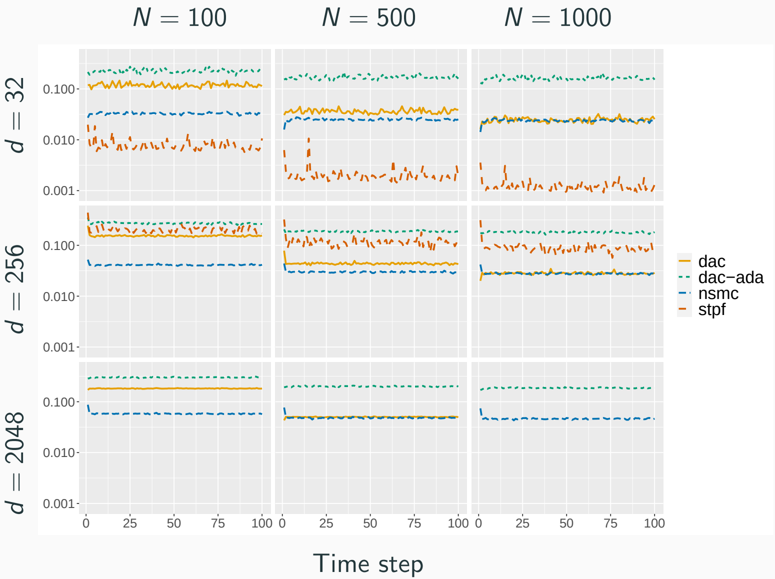
$$
\begin{aligned}
f_1(x_1) &= \mathcal{N}(x_1; m_1, \Sigma_1) \\
f_t(x_{t-1}, x_t) &= \mathcal{N}(x_t; Ax_{t-1}, \Sigma) \\
g_t(x_t, y_t) &= \mathcal{N}(y_t; x_t, \sigma_y^2 Id_d),
\end{aligned}
$$

with $\Sigma \in \mathbb{R}^{d \times d}$ a tridiagonal matrix.

We compare the RMSE

$$
\mathrm{RMSE}(x_t(i)) := \frac{\mathbb{E}\left[(\bar{x}_t(i) - \mu_{t,i})^2\right]}{\sigma_{t,i}^2}
$$

Linear Gaussian Model - RMSE



Spatial Model

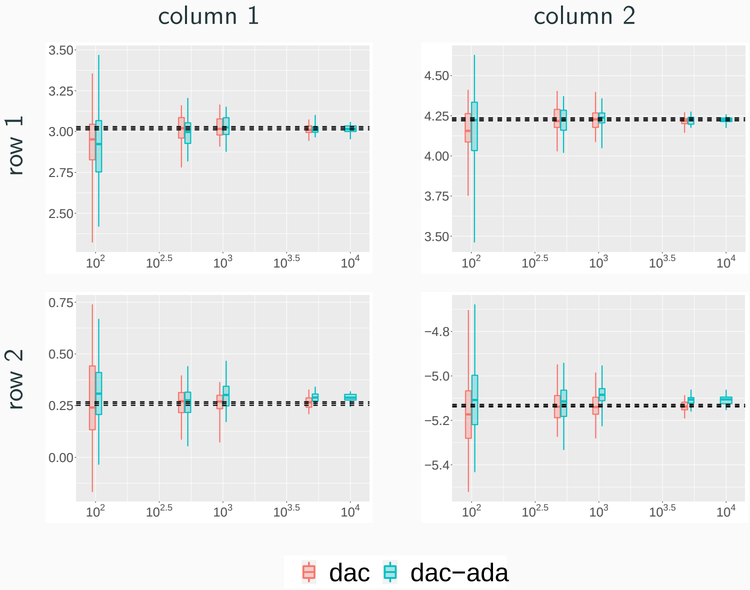
The components of X_t are indexed by the vertices $v \in V$ of a lattice, where $V = \{1, \ldots, d\}^2$, and

$$X_t(v) = X_{t-1}(v) + U_t(v), \qquad U_t(v) \sim \mathcal{N}(0, \sigma_x^2)$$

and

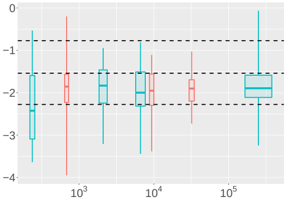
$$g_t(x_t, y_t) \propto \left[1 + \nu^{-1} \sum_{v \in V} \left((y_t(v) - x_t(v)) \sum_{j : D(v,j) \le r_y} \tau^{D(v,j)} (y_t(j) - x_t(j)) \right) \right]^{-(\nu + d^2)/2}.$$

2×2 grid



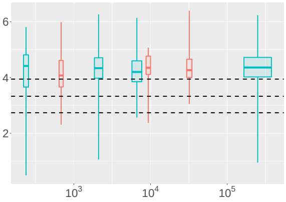
8×8 grid

$(1, 1)$



Runtime / s

$(8, 6)$



Runtime / s

Multivariate Stochastic Volatility Model

$$
\begin{aligned}
X_1 &\sim \mathcal{N}_d(0, \Sigma_0), \\
Y_t &= C_t^{1/2} V_t, \\
X_{t+1} &= \Phi X_t + U_t,
\end{aligned}
$$

with C_t a diagonal matrix with entries

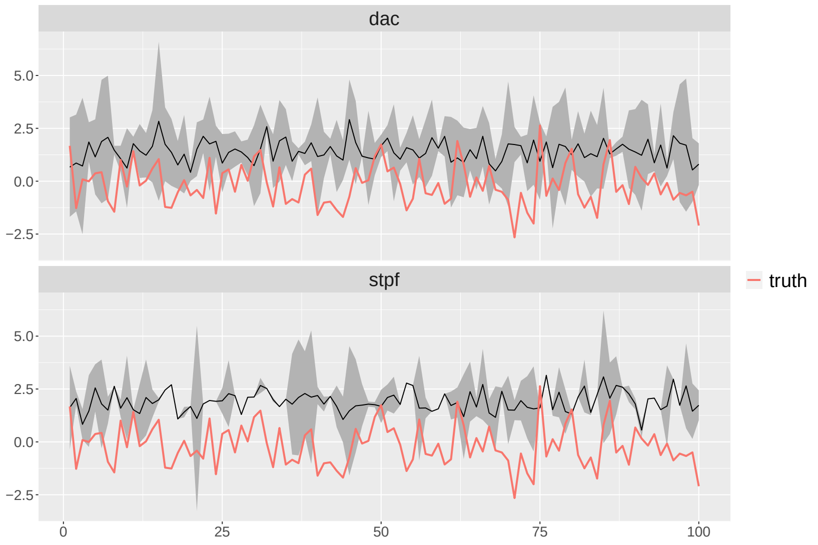
$$
C_t = \operatorname{diag}\left(\exp(X_t(1)), \exp(X_t(2)), \ldots, \exp(X_t(d))\right),
$$

Φ a diagonal matrix with diagonal entries ϕ_i,

$$
\begin{pmatrix} V_t \\ U_t \end{pmatrix} \sim \mathcal{N}_{2d}(0, \Sigma) \qquad \text{with } \Sigma = \begin{pmatrix} \Sigma_{VV} & \Sigma_{VU} \\ \Sigma_{VU} & \Sigma_{UU} \end{pmatrix},
$$

and Σ_0 has entries given by

$$
(\Sigma_0)_{ij} = (\Sigma_{UU})_{ij} / (1 - \phi_i \phi_j),
$$


dac
stpf
5.0
2.5
0.0
−2.5
0
25
50
75
100
truth

Opportunities and Related Ideas

- Theoretical guarantees should follow from those of standard DaC-SMC (Kuntz et al., 2021) and marginal particle filters (Crucinio and Johansen, 2023)
- smoothing, parameter estimation
- Other applications: Parallel (in time) Smoothing (Corneflos et al., 2022; Ding and Gandy, 2018), Divide and Conquer Fusion (Chan et al., 2021), inference for graphical models and phylogenetic trees (Jewell, 2015; Paige and Wood, 2016; Lindsten et al., 2017)

Thank you!

Bibliography i

Frédéric Cérou, Pierre Del Moral, Teddy Furon, and Arnaud Guyader. Sequential monte carlo for rare event estimation. *Statistics and computing*, 22(3):795–808, 2012.

R. Chan, M. Pollock, A. M. Johansen, and G. O. Roberts. Divide-and-conquer Monte Carlo fusion. *arXiv preprint arXiv:2110.07265*, 2021.

N Chopin and O Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer, Cham, 2020. doi: 10.1007/978-3-030-47845-2.

A. Corneflos, N. Chopin, and S. Särkkä. De-Sequentialized Monte Carlo: a parallel-in-time particle smoother. *Journal of Machine Learning Research*, 23(283):1–39, 2022.

F. R. Crucinio and A. M. Johansen. Properties of marginal sequential Monte Carlo methods. *arXiv preprint arXiv:2303.03498*, 2023.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

D. Ding and A. Gandy. Tree-based particle smoothing algorithms in a hidden markov model. *arXiv preprint arXiv:1808.08400*, 2018.

S. W. Jewell. *Divide and conquer sequential Monte Carlo for phylogenetics*. Master thesis, University of British Columbia, 2015.

Mike Klaas, Nando De Freitas, and Arnaud Doucet. Toward practical N^2 Monte Carlo: The marginal particle filter. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence*, pages 308–315, 2005.

J. Kuntz, F. R. Crucinio, and A. M. Johansen. The divide-and-conquer sequential Monte Carlo algorithm: theoretical properties and limit theorems. *arXiv preprint arXiv:2110.15782*, 2021.

Juan Kuntz, Francesca R Crucinio, and Adam M Johansen. Product-form estimators: exploiting independence to scale up Monte Carlo. *Statistics and Computing*, 32(1):1–22, 2022.

Fredrik Lindsten, Adam M Johansen, Christian A Næsseth, Bonnie Kirkpatrick, Thomas B Schön, John A D Aston, and Alexandre Bouchard-Côté. Divide-and-Conquer with sequential Monte Carlo. *Journal of Computational and Graphical Statistics*, 26(2):445–458, 2017.

Andrew A Manderson and Robert JB Goudie. Combining chains of bayesian models with markov melding. *arXiv preprint arXiv:2111.11566*, 2021.

B. Paige and F. Wood. Inference networks for sequential monte carlo in graphical models. In *33rd Int Conf Mach Learn*, volume 48, pages 3040–3049, 2016.

S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential Monte Carlo without likelihoods. 104(4):1760–1765, February 2007.