Nested Sequential Monte Carlo Methods

Christian A. Naesseth CHRISTIAN.A.NAESSETH@LIU.SE
Linköping University, Linköping, Sweden

Fredrik Lindsten FREDRIK.LINDSTEN@ENG.CAM.AC.UK
The University of Cambridge, Cambridge, United Kingdom

Thomas B. Schön THOMAS.SCHON@IT.UU.SE
Uppsala University, Uppsala, Sweden

Abstract

We propose *nested sequential Monte Carlo* (NSMC), a methodology to sample from sequences of probability distributions, even where the random variables are high-dimensional. NSMC generalises the SMC framework by requiring only approximate, *properly weighted*, samples from the SMC proposal distribution, while still resulting in a correct SMC algorithm. Furthermore, NSMC can in itself be used to produce such properly weighted samples. Consequently, one NSMC sampler can be used to construct an efficient high-dimensional proposal distribution for another NSMC sampler, and this *nesting* of the algorithm can be done to an arbitrary degree. This allows us to consider complex and high-dimensional models using SMC. We show results that motivate the efficacy of our approach on several filtering problems with dimensions in the order of 100 to 1 000.

1. Introduction

Inference in complex and high-dimensional statistical models is a very challenging problem that is ubiquitous in applications. Examples include, but are definitely not limited to, climate informatics (Monteleoni et al., 2013), bioinformatics (Cohen, 2004) and machine learning (Wainwright & Jordan, 2008). In particular, we are interested in *sequential* Bayesian inference, which involves computing integrals of the form

$$\bar{\pi}_k(f) := \mathbb{E}_{\bar{\pi}_k}[f(X_{1:k})] = \int f(x_{1:k})\bar{\pi}_k(x_{1:k})\mathrm{d}x_{1:k}, \quad (1)$$

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

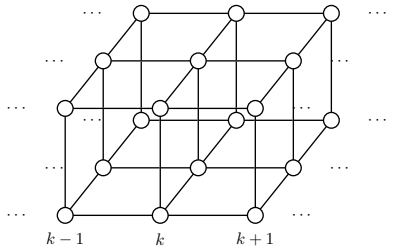

Figure 1. Example of a spatio-temporal model where $\bar{\pi}_k(x_{1:k})$ is given by a $k \times 2 \times 3$ undirected graphical model and $x_k \in \mathbb{R}^{2\times 3}$.

for some sequence of probability densities

$$\bar{\pi}_k(x_{1:k}) = Z_{\pi_k}^{-1}\pi_k(x_{1:k}), \qquad k \geq 1, \quad (2)$$

with normalisation constants $Z_{\pi_k} = \int \pi_k(x_{1:k})\mathrm{d}x_{1:k}$. Note that $x_{1:k} := (x_1, \ldots, x_k) \in \mathsf{X}_k$. The typical scenario that we consider is the well-known problem of inference in time series or state space models (Shumway & Stoffer, 2011; Cappé et al., 2005). Here the index k corresponds to time and we want to process some *observations* $y_{1:k}$ in a sequential manner to compute expectations with respect to the filtering distribution $\bar{\pi}_k(\mathrm{d}x_k) = \mathbb{P}(X_k \in \mathrm{d}x_k \,|\, y_{1:k})$. To be specific, we are interested in settings where

(i) X_k is high-dimensional, i.e. $X_k \in \mathbb{R}^d$ with $d \gg 1$, and

(ii) there are *local dependencies* among the latent variables $X_{1:k}$, both w.r.t. time k and between the individual components of the (high-dimensional) vectors X_k.

One example of the type of models we consider are the so-called spatio-temporal models (Wikle, 2015; Cressie & Wikle, 2011; Rue & Held, 2005). In Figure 1 we provide a probabilistic graphical model representation of a spatio-temporal model that we will explore further in Section 6.

Sequential Monte Carlo (SMC) methods, reviewed in Section 2.1, comprise one of the most successful methodolo-

gies for sequential Bayesian inference. However, SMC struggles in high-dimensions and these methods are rarely used for dimensions, say, $d \geq 10$ (Rebeschini & van Handel, 2015). The purpose of the NSMC methodology is to push this limit well beyond $d = 10$.

The basic strategy, described in Section 2.2, is to mimic the behaviour of a so-called *fully adapted* SMC algorithm. Full adaptation can drastically improve the efficiency of SMC in high dimensions. Unfortunately, it can rarely be implemented in practice since the fully adapted proposal distributions are typically intractable. NSMC addresses this difficulty by requiring only approximate, *properly weighted*, samples from the proposal distribution. The proper weighting condition ensures the validity of NSMC, thus providing a generalisation of the family of SMC methods. Furthermore, NSMC will itself produce properly weighted samples. Consequently, it is possible to use one NSMC procedure within another to construct efficient high-dimensional proposal distributions. This *nesting* of the algorithm can be done to an arbitrary degree. For instance, for the model depicted in Figure 1 we could use three nested samplers, one for each dimension of the "volume".

The main methodological development is concentrated to Sections 3–4. We introduce the concept of proper weighting, approximations of the proposal distribution, and nesting of Monte Carlo algorithms. Throughout Section 3 we consider simple importance sampling and in Section 4 we extend the development to the sequential setting.

We deliberately defer the discussion of the existing body of related work until Section 5, to open up for a better understanding of the relationships to the new developments presented in Sections 3–4. We also discuss various attractive features of NSMC that are of interest in high-dimensional settings, e.g. the fact that it is easy to distribute the computation, which results in improved memory efficiency and lower communication costs. Finally, Section 6 profiles our method extensively with a state-of-the-art competing algorithm on several high-dimensional data sets. We also show the performance of inference and the modularity of the method on a $d = 1\,056$ dimensional climatological spatio-temporal model (Fu et al., 2012) structured according to Figure 1.

2. Background and Inference Strategy

2.1. Sequential Monte Carlo

Evaluating $\bar{\pi}_k(f)$ as well as the normalisation constant Z_{π_k} in (2) is typically intractable and we need to resort to approximations. SMC methods, or particle filters (PF), constitute a popular class of numerical approximations for sequential inference problems. Here we give a high-level introduction to the concepts underlying SMC methods, and postpone the details to Section 4. For a more extensive treatment we refer to Doucet & Johansen (2011); Cappé et al. (2005); Doucet et al. (2001). In particular, we will use the auxiliary SMC method as proposed by Pitt & Shephard (1999).

At iteration $k-1$, the SMC sampler approximates the target distribution $\bar{\pi}_{k-1}$ by a collection of weighted *particles* $\{(X^i_{1:k-1}, W^i_{k-1})\}_{i=1}^N$. These samples define an empirical point-mass approximation of the target distribution

$$\bar{\pi}^N_{k-1}(\mathrm{d}x_{1:k-1}) := \sum_{i=1}^{N} \frac{W^i_{k-1}}{\sum_\ell W^\ell_{k-1}} \delta_{X^i_{1:k-1}}(\mathrm{d}x_{1:k-1}), \quad (3)$$

where $\delta_X(\mathrm{d}x)$ denotes a Dirac measure at X. Each iteration of the SMC method can then conceptually be described by three steps, resampling, propagation, and weighting.

The resampling step puts emphasis on the most promising particles by discarding the unlikely ones and duplicating the likely ones. The propagation and weighting steps essentially correspond to using importance sampling when changing the target distribution from $\bar{\pi}_{k-1}$ to $\bar{\pi}_k$, i.e. simulating new particles from a *proposal distribution* and then computing corresponding importance weights.

2.2. Adapting the Proposal Distribution

The first working SMC algorithm was the bootstrap PF by Gordon et al. (1993), which propagates particles by sampling from the system dynamics and computes importance weights according to the observation likelihood (in the state space setting). However, it is well known that the bootstrap PF suffers from weight collapse in high-dimensional settings (Bickel et al., 2008), i.e. the estimate is dominated by a single particle with weight close to one. This is an effect of the mismatch between the importance sampling proposal and the target distribution, which typically gets more pronounced in high dimensions.

More efficient proposals, partially alleviating the degeneracy issue for some models, can be designed by *adapting* the proposal distribution to the target distribution (see Section 4.2). In Naesseth et al. (2014a) we make use of the *fully adapted* SMC method (Pitt & Shephard, 1999) for doing inference in a (fairly) high-dimensional *discrete* model where x_k is a 60-dimensional discrete vector. We can then make use of forward filtering and backward simulation, operating on the individual *components* of each x_k, in order to sample from the fully adapted SMC proposals. However, this method is limited to models where the latent space is either discrete or Gaussian and the optimal proposal can be identified with a tree-structured graphical model. Our development here can be seen as a non-trivial extension of this technique. Instead of coupling one SMC sampler with an *exact* forward filter/backward simulator (which in fact

reduces to an instance of standard SMC), we derive a way of coupling multiple SMC samplers and SMC-based backward simulators. This allows us to construct procedures for mimicking the efficient fully adapted proposals for arbitrary latent spaces and structures in high-dimensional models.

3. Proper Weighting and Nested Importance Sampling

In this section we will lay the groundwork for the derivation of the class of NSMC algorithms. We start by considering the simpler case of importance sampling (IS), which is a fundamental component of SMC, and introduce the key concepts that we make use of. In particular, we will use a (slightly nonstandard) presentation of an algorithm as an instance of a *class*, in the object-oriented sense, and show that these classes can be nested to an arbitrary degree.

3.1. Exact Approximation of the Proposal Distribution

Let $\bar{\pi}(x) = Z_\pi^{-1}\pi(x)$ be a target distribution of interest. IS can be used to estimate an expectation $\bar{\pi}(f) := \mathbb{E}_{\bar{\pi}}[f(X)]$ by sampling from a proposal distribution $\bar{q}(x) = Z_q^{-1}q(x)$ and computing the estimator $(\sum_{i=1}^N W^i)^{-1}\sum_{i=1}^N W^i f(X^i)$, with $W^i = \frac{Z_q\pi(X^i)}{q(X^i)}$, and where $\{(X^i, W^i)\}_{i=1}^N$ are the weighted samples. It is possible to replace the IS weight by a nonnegative unbiased estimate, and still obtain a valid (consistent, etc.) algorithm (Liu, 2001, p. 37). One way to motivate this approach is by considering the random weight to be an auxiliary variable and to extend the target distribution accordingly. Our development is in the same flavour, but we will use a more explicit condition on the relationship between the random weights and the simulated particles. Specifically, we will make use of the following key property to formally justify the proposed algorithms.

Definition 1 (Properly weighted sample)**.** *A (random) pair (X, W) is properly weighted for an* unnormalised *distribution p if $W \geq 0$ and $\mathbb{E}[f(X)W] = p(f) := \int f(x)p(x)\mathrm{d}x$ for all measurable functions f.*

Note that proper weighting of $\{(X^i, W^i)\}_{i=1}^N$ implies unbiasedness of the estimate of the normalising constant of p. Indeed, taking $f(x) \equiv 1$ gives $\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N W^i\right] = \int p(x)\mathrm{d}x =: Z_p$.

Interestingly, to construct a valid IS algorithm for our target $\bar{\pi}$ it is sufficient to generate samples that are properly weighted *w.r.t. the proposal* distribution q. To formalise this claim, assume that we are not able to simulate exactly from $\bar{q}$, but that it is possible to evaluate the unnormalised density q point-wise. Furthermore, assume we have access to a class Q, which works as follows. The constructor of Q requires the specification of an *unnormalised density function*, say, q, which will be approximated by the procedures of Q. Furthermore, to highlight the fact that we will typically use IS (and SMC) to construct Q, the constructor also takes as an argument a precision parameter M, corresponding to the number of samples used by the "internal" Monte Carlo procedure. An object is then instantiated as $\mathsf{q} = \mathsf{Q}(q, M)$. The class Q is assumed to have the following properties:

(A1) Let $\mathsf{q} = \mathsf{Q}(q, M)$. Assume that:

1. The construction of q results in the generation of a (possibly random) member variable, accessible as $\widehat{Z}_q = \mathsf{q}.\mathsf{GetZ}()$. The variable $\widehat{Z}_q$ is a nonnegative, unbiased estimate of the normalising constant $Z_q = \int q(x)dx$.
2. Q has a member function $\mathsf{Simulate}$ which returns a (possibly random) variable $X = \mathsf{q}.\mathsf{Simulate}()$, such that $(X, \widehat{Z}_q)$ is properly weighted for q.

With the definition of Q in place, it is possible to generalise[1] the basic importance sampler as in Algorithm 1, which generates weighted samples $\{(X^i, W^i)\}_{i=1}^N$ targeting $\bar{\pi}$. Note that Algorithm 1 is different from a random weight IS, since it approximates the proposal distribution (and not just the importance weights).

Algorithm 1 Nested IS (steps 1–3 for $i = 1, \ldots, N$)

1. Initialise $\mathsf{q}^i = \mathsf{Q}(q, M)$.
2. Set $\widehat{Z}_q^i = \mathsf{q}^i.\mathsf{GetZ}()$ and $X^i = \mathsf{q}^i.\mathsf{Simulate}()$.
3. Set $W^i = \dfrac{\widehat{Z}_q^i \pi(X^i)}{q(X^i)}$.
4. Compute $\widehat{Z}_\pi = \frac{1}{N}\sum_{i=1}^N W^i$.

To see the validity of Algorithm 1 we can interpret the sampler as a standard IS algorithm for an extended target distribution, defined as $\bar{\Pi}(x, u) := u\,\bar{Q}(x, u)\bar{\pi}(x)q^{-1}(x)$, where $\bar{Q}(x, u)$ is the joint PDF of the random pair $(\mathsf{q}.\mathsf{Simulate}(), \mathsf{q}.\mathsf{GetZ}())$. Note that $\bar{\Pi}$ is indeed a PDF that admits $\bar{\pi}$ as a marginal; for any measurable subset $A \subseteq \mathsf{X}$,

$$\bar{\Pi}(A \times \mathbb{R}_+) = \int \mathbb{1}_A(x)\frac{u\,\bar{\pi}(x)}{q(x)}\bar{Q}(x, u)\mathrm{d}x\mathrm{d}u$$
$$= \mathbb{E}\left[\widehat{Z}_q \frac{\mathbb{1}_A(X)\bar{\pi}(X)}{q(X)}\right] = \bar{q}\left(\mathbb{1}_A\frac{\bar{\pi}}{q}\right)Z_q = \bar{\pi}(A),$$

where the penultimate equality follows from the fact that $(X, \widehat{Z}_q)$ is properly weighted for q. Furthermore, the standard unnormalised IS weight for a sampler with target $\bar{\Pi}$

[1] With $\mathsf{q}.\mathsf{GetZ}() \mapsto Z$ and $\mathsf{q}.\mathsf{Simulate}()$ returning a sample from $\bar{q}$ we obtain the standard IS method.

and proposal $\bar{Q}$ is given by $u\,\pi/q$, in agreement with Algorithm 1.

Algorithm 1 is an example of what is referred to as an *exact approximation*; see e.g., Andrieu & Roberts (2009); Andrieu et al. (2010). Algorithmically, the method appears to be an approximation of an IS, but samples generated by the algorithm nevertheless target the correct distribution $\bar{\pi}$.

3.2. Modularity of Nested IS

To be able to implement Algorithm 1 we need to define a class Q with the required properties (A1). The modularity of the procedure (as well as its name) comes from the fact that we can use Algorithm 1 also in this respect. Indeed, let us now view $\bar{\pi}$—the target distribution of Algorithm 1—as the *proposal distribution* for another Nested IS procedure and consider the following definition of Q:

1. Algorithm 1 is executed at the construction of the object $\mathsf{p} = \mathsf{Q}(\pi, N)$, and $\mathsf{p.GetZ}()$ returns the normalising constant estimate $\widehat{Z}_\pi$.
2. $\mathsf{p.Simulate}()$ simulates a categorical random variable B with $\mathbb{P}(B = i) = W^i / \sum_{\ell=1}^N W^\ell$ and returns X^B.

A simple computation now yields that for any measurable f we have $\mathbb{E}[f(X^B)\widehat{Z}_\pi] = \bar{\pi}(f)Z_\pi$. This implies that $(X^B, \widehat{Z}_\pi)$ is properly weighted for π and that our definition of $\mathsf{Q}(\pi, N)$ indeed satisfies condition (A1).

The Nested IS algorithm in itself is unlikely to be of direct practical interest. However, in the next section we will, essentially, repeat the preceding derivation in the context of SMC to develop the NSMC method.

4. Nested Sequential Monte Carlo

4.1. Fully Adapted SMC Samplers

Let us return to the sequential inference problem. As before, let $\bar{\pi}_k(x_{1:k}) = Z_{\pi_k}^{-1}\pi_k(x_{1:k})$ denote the target distribution at "time" k. The unnormalised density π_k can be evaluated point-wise, but the normalising constant Z_{π_k} is typically unknown. We will use SMC to simulate sequentially from the distributions $\{\bar{\pi}_k\}_{k=1}^n$. In particular, we consider the fully adapted SMC sampler (Pitt & Shephard, 1999), which corresponds to a specific choice of resampling weights and proposal distribution, chosen in such a way that the importance weights are all equal to $1/N$. Specifically, the proposal distribution (often referred to as the *optimal proposal*) is given by $\bar{q}_k(x_k \,|\, x_{1:k-1}) = Z_{q_k}(x_{1:k-1})^{-1} q_k(x_k \,|\, x_{1:k-1})$, where

$$q_k(x_k \,|\, x_{1:k-1}) := \pi_k(x_{1:k})/\pi_{k-1}(x_{1:k-1}).$$

In addition, the normalising "constant" $Z_{q_k}(x_{1:k-1}) = \int q_k(x_k \,|\, x_{1:k-1})\mathrm{d}x_k$ is further used to define the *resampling weights*, i.e. the particles at time $k-1$ are resampled according to $Z_{q_k}(x_{1:k-1})$ before they are propagated to time k. For notational simplicity, we use the convention $x_{1:0} = \emptyset$, $q_1(x_1 \,|\, x_{1:0}) = \pi_1(x_1)$ and $Z_{q_1}(x_{1:0}) = Z_{\pi_1}$. The fully adapted auxiliary SMC sampler is given in Algorithm 2.

Algorithm 2 SMC (fully adapted)

1. Set $\widehat{Z}_{\pi_0} = 1$.
2. **for** $k = 1$ **to** n
   (a) Compute $\widehat{Z}_{\pi_k} = \widehat{Z}_{\pi_{k-1}} \times \frac{1}{N}\sum_{j=1}^N Z_{q_k}(X_{1:k-1}^j)$.
   (b) Draw $m_k^{1:N}$ from a multinomial distribution with probabilities $\frac{Z_{q_k}(X_{1:k-1}^j)}{\sum_{\ell=1}^N Z_{q_k}(X_{1:k-1}^\ell)}$, for $j = 1, \ldots, N$.
   (c) Set $L \leftarrow 0$
   (d) **for** $j = 1$ **to** N
      i. Draw $X_k^i \sim \bar{q}_k(\cdot \,|\, X_{1:k-1}^j)$ and let $X_{1:k}^i = (X_{1:k-1}^j, X_k^i)$ for $i = L+1, \ldots, L+m_k^j$.
      ii. Set $L \leftarrow L + m_k^j$.

As mentioned above, at each iteration $k = 1, \ldots, n$, the method produces *unweighted* samples $\{X_k^i\}_{i=1}^N$ approximating $\bar{\pi}_k$. It also produces an unbiased estimate $\widehat{Z}_{\pi_k}$ of Z_{π_k} (Del Moral, 2004, Proposition 7.4.1). The algorithm is expressed in a slightly non-standard form; at iteration k we loop over the ancestor particles, i.e. the particles after resampling at iteration $k-1$, and let each ancestor particle j generate m_k^j offsprings. (The variable L is just for bookkeeping.) This is done to clarify the connection with the NSMC procedure below. Furthermore, we have included a (completely superfluous) resampling step at iteration $k = 1$, where the "dummy variables" $\{X_{1:0}^i\}_{i=1}^N$ are resampled according to the (all equal) weights $\{Z_{q_1}(X_{1:0}^i)\}_{i=1}^N = \{Z_{\pi_1}\}_{i=1}^N$. The analogue of this step is, however, used in the NSMC algorithm, where the initial normalising constant Z_{π_1} is *estimated*. We thus have to resample the corresponding initial particle systems accordingly.

4.2. Fully Adapted Nested SMC Samplers

In analogue with Section 3, assume now that we are not able to simulate exactly from $\bar{q}_k$, nor compute Z_{q_k}. Instead, we have access to a class Q which satisfies condition (A1). The proposed NSMC method is then given by Algorithm 3.

Algorithm 3 can be seen as an *exact approximation* of the fully adapted SMC sampler in Algorithm 2. (In Naesseth et al. (2015) we provide a formulation of NSMC with arbitrary proposals and resampling weights.) We replace the exact computation of Z_{q_k} and exact simulation from $\bar{q}_k$, by the approximate procedures available through Q. Despite

Algorithm 3 Nested SMC (fully adapted)

1. Set $\widehat{Z}_{\pi_0} = 1$.
2. **for** $k = 1$ **to** n
   (a) Initialise $\mathsf{q}^j = \mathsf{Q}(q_k(\cdot \mid X_{1:k-1}^j), M)$ for $j = 1, \ldots, N$.
   (b) Set $\widehat{Z}_{q_k}^j = \mathsf{q}^j.\mathsf{GetZ}()$ for $j = 1, \ldots, N$.
   (c) Compute $\widehat{Z}_{\pi_k} = \widehat{Z}_{\pi_{k-1}} \times \left\{ \frac{1}{N} \sum_{j=1}^N \widehat{Z}_{q_k}^j \right\}$.
   (d) Draw $m_k^{1:N}$ from a multinomial distribution with probabilities $\frac{\widehat{Z}_{q_k}^j}{\sum_{\ell=1}^N \widehat{Z}_{q_k}^\ell}$ for $j = 1, \ldots, N$.
   (e) Set $L \leftarrow 0$
   (f) **for** $j = 1$ **to** N
      i. Compute $X_k^i = \mathsf{q}^j.\mathsf{Simulate}()$ and let $X_{1:k}^i = (X_{1:k-1}^j, X_k^i)$ for $i = L+1, \ldots, L+m_k^j$.
      ii. **delete** q^j.
      iii. Set $L \leftarrow L + m_k^j$.

this approximation, however, Algorithm 3 is a valid SMC method. This is formalised by the following theorem.

Theorem 1. *Assume that* Q *satisfies condition (A1). Then, under certain regularity conditions on the function* $f : \mathsf{X}_k \mapsto \mathbb{R}^d$ *and for an asymptotic variance* $\Sigma_k^M(f)$*, both specified in Naesseth et al. (2015), we have*

$$N^{1/2} \left(\frac{1}{N} \sum_{i=1}^N f(X_{1:k}^i) - \bar{\pi}_k(f) \right) \xrightarrow{D} \mathcal{N}(0, \Sigma_k^M(f)),$$

where $\{X_{1:k}^i\}_{i=1}^M$ *are generated by Algorithm 3 and* $\xrightarrow{D}$ *denotes convergence in distribution.*

Proof. See Naesseth et al. (2015). □

Remark 1. The key point with Theorem 1 is that, under certain regularity conditions, the NSMC method converges at rate $\sqrt{N}$ even for a fixed (and finite) value of the precision parameter M. The asymptotic variance $\Sigma_k^M(f)$, however, will depend on the accuracy and properties of the approximative procedures of Q. We leave it as future work to establish more informative results, relating the asymptotic variance of NSMC to that of the ideal, fully adapted SMC sampler.

4.3. Backward Simulation and Modularity of NSMC

As previously mentioned, the NSMC procedure is modular in the sense that we can make use of Algorithm 3 also to define the class Q. Thus, we now view $\bar{\pi}_n$ as the *proposal distribution* that we wish to approximately sample from using NSMC. Algorithm 3 directly generates an estimate $\widehat{Z}_{\pi_n}$ of the normalising constant of π_n (which indeed is unbiased, see Theorem 2). However, we also need to generate a sample $\widetilde{X}_{1:n}$ such that $(\widetilde{X}_{1:n}, \widehat{Z}_{\pi_n})$ is properly weighted for π_n.

The simplest approach, akin to the Nested IS procedure described in Section 3.2, is to draw B_n uniformly on $\{1, \ldots, N\}$ and return $\widetilde{X}_{1:n} = X_{1:n}^{B_n}$. This will indeed result in a valid definition of the $\mathsf{Simulate}$ procedure. However, this approach will suffer from the well known path degeneracy of SMC samplers. In particular, since we call $\mathsf{q}^j.\mathsf{Simulate}()$ multiple times in Step 2(f)i of Algorithm 3, we risk to obtain (very) strongly correlated samples by this simple approach.

It is possible to improve the performance of the above procedure by instead making use of a *backward simulator* (Godsill et al., 2004; Lindsten & Schön, 2013) to simulate $\widetilde{X}_{1:n}$. The backward simulator, given in Algorithm 4, is a type of smoothing algorithm; it makes use of the particles generated by a forward pass of Algorithm 3 to simulate backward in "time" a trajectory $\widetilde{X}_{1:n}$ approximately distributed according to $\bar{\pi}_n$.

Algorithm 4 Backward simulator (fully adapted)

1. Draw B_n uniformly on $\{1, \ldots, N\}$.
2. Set $\widetilde{X}_n = X_n^{B_n}$.
3. **for** $k = n-1$ **to** 1
   (a) Compute $\widetilde{W}_k^j = \frac{\pi_n((X_{1:k}^j, \widetilde{X}_{k+1:n}))}{\pi_k(X_{1:k}^j)}$ for $j = 1, \ldots, N$.
   (b) Draw B_k from a categorical distribution with probabilities $\frac{\widetilde{W}_k^j}{\sum_{\ell=1}^N \widetilde{W}_k^\ell}$ for $j = 1, \ldots, N$.
   (c) Set $\widetilde{X}_{k:n} = (X_k^{B_k}, \widetilde{X}_{k+1:n})$.

Remark 2. Algorithm 4 assumes unweighted particles and can thus be used in conjunction with the fully adapted NSMC procedure of Algorithm 2. If, however, the forward filter is not fully adapted the weights need to be accounted for in the backward simulation; see Naesseth et al. (2015).

The modularity of NSMC is established by the following result.

Definition 2. *Let* $\mathsf{p} = \mathsf{Q}(\pi_n, N)$ *be defined as follows:*

1. *The constructor executes Algorithm 3 with target distribution* π_n *and with* N *particles, and* $\mathsf{p}.\mathsf{GetZ}()$ *returns the estimate of the normalising constant* $\widehat{Z}_{\pi_n}$.
2. $\mathsf{p}.\mathsf{Simulate}()$ *executes Algorithm 4 and returns* $\widetilde{X}_{1:n}$.

Theorem 2. *The class* Q *defined as in Definition 2 satisfies condition (A1).*

Proof. See Naesseth et al. (2015). □

A direct, and important, consequence of Theorem 2 is that NSMC can be used as a component of powerful learning algorithms, such as the particle Markov chain Monte Carlo (PMCMC) method (Andrieu et al., 2010) and many of the other methods discussed in Section 5. Since standard SMC is a special case of NSMC, Theorem 2 implies proper weighting also of SMC.

5. Practicalities and Related Work

There has been much recent interest in using SMC within SMC in various ways. The SMC^2 by Chopin et al. (2013) and the recent method by Crisan & Míguez (2013) are sequential learning algorithms for state space models, where one SMC sampler for the parameters is coupled with another SMC sampler for the latent states. Johansen et al. (2012) and Chen et al. (2011) address the state inference problem by splitting the state variable into different components and run coupled SMC samplers for these components. These methods differ substantially from NSMC; they solve different problems and the "internal" SMC sampler(s) is constructed in a different way (for approximate marginalisation instead of for approximate simulation). Another related method is the random weights PF of Fearnhead et al. (2010a), requiring exact samples from $\bar{q}$ and where the importance weights are estimated using a nested Monte Carlo algorithm.

The method most closely related to NSMC is the space-time particle filter (ST-PF) (Beskos et al., 2014a), which has been developed independently and in parallel with our work. The ST-PF is also designed for solving inference problems in high-dimensional models. It can be seen as a island PF (Vergé et al., 2013) implementation of the method presented by Naesseth et al. (2014b). Specifically, for a spatio-temporal models they run an island PF over both spatial and temporal dimensions. However, the ST-PF does not generate an approximation of the fully adapted SMC sampler.

Another key distinction between NSMC and ST-PF is that in the latter each particle in the "outer" SMC sampler comprises a complete particle system from the "inner" SMC sampler. For NSMC, on the other hand, the particles will simply correspond to different hypotheses about the latent variables (as in standard SMC), regardless of how many samplers that are nested. This is a key feature of NSMC, since it implies that it is easily distributed over the particles. The main computational effort of Algorithm 3 is the construction of $\{\mathsf{q}^j\}_{j=1}^N$ and the calls to the Simulate procedure, which can be done independently for each particle. This leads to improved memory efficiency and lower communication costs. Furthermore, we have found (see Section 6) that NSMC can outperform ST-PF even when run on a single machine with matched computational costs.

Another strength of NSMC methods are their relative ease of implementation, which we show in Section 6.3. We use the framework to sample from what is essentially a cubic grid Markov random field (MRF) model just by implementing three nested samplers, each with a target distribution defined on a simple chain.

There are also other SMC-based methods designed for high-dimensional problems, e.g., the block PF studied by Rebeschini & van Handel (2015), the location particle smoother by Briggs et al. (2013) and the PF-based methods reviewed in Djuric & Bugallo (2013). However, these methods are all inconsistent, as they are based on various approximations that result in systematic errors.

The previously mentioned PMCMC (Andrieu et al., 2010) is a related method, where SMC is used as a component of an MCMC algorithm. We make use of a very similar extended space approach to motivate the validity of our algorithm. Note that our proposed algorithm can be used as a component in PMCMC and most of the other algorithms mentioned above, which further increases the scope of models it can handle.

6. Experimental Results

We illustrate NSMC on three high-dimensional examples, both with real and synthetic data. We compare NSMC with standard (bootstrap) PF and the ST-PF of Beskos et al. (2014a) with equal computational budgets on a single machine (i.e., neglecting the fact that NSMC is more easily distributed). These methods are, to the best of our knowledge, the only other available *consistent* online methods for full Bayesian inference in general sequential models. For more detailed explanations of the models and additional results, see Naesseth et al. (2015)[2].

6.1. Gaussian State Space Model

We start by considering a high-dimensional Gaussian state space model, where we have access to the true solution from the Kalman filter (Kalman, 1960). The latent variables and measurements $\{X_{1:k}, Y_{1:k}\}$, with $\{X_k, Y_k\} = \{X_{k,l}, Y_{k,l}\}_{l=1}^d$, are modeled by a $d \times k$ lattice Gaussian MRF, which can be identified with a linear Gaussian state space model (see Naesseth et al. (2015)). We run a 2-level NSMC sampler. The outer level is fully adapted, i.e. the proposal distribution is $q_k = p(x_k \mid x_{k-1}, y_k)$, which thus constitute the target distribution for the inner level. To generate properly weighted samples from q_k, we use a bootstrap PF operating on the d components of the vector x_k. Note that we only use bootstrap proposals where the actual sampling takes place, and that the conditional distribution

[2]Code available at `https://github.com/can-cs/nestedsmc`

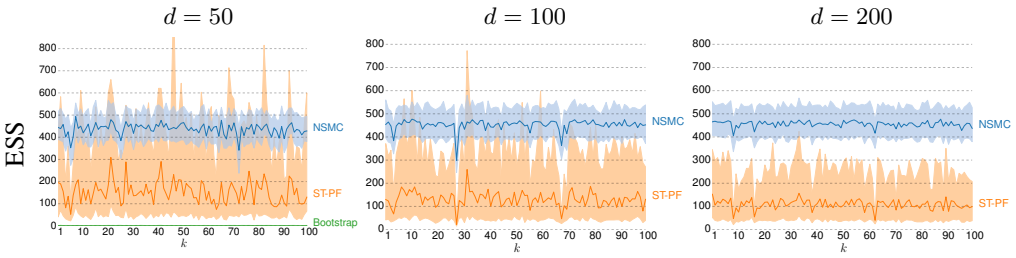

Figure 2. Median (over dimension) ESS (4) and 15–85% percentiles (shaded region). The results are based on 100 independent runs for the Gaussian MRF with dimension d.

$p(x_k \mid x_{k-1}, y_k)$ is not explicitly used.

We simulate data from this model for $k = 1, \ldots, 100$ for different values of $d = \dim(x_k) \in \{50, 100, 200\}$. The exact filtering marginals are computed using the Kalman filter. We compare with both the ST-PF and standard (bootstrap) PF.

The results are evaluated based on the effective sample size (ESS, see e.g. Fearnhead et al. (2010b)) defined as,

$$\mathrm{ESS}(x_{k,l}) = \left(\mathbb{E}\left[\frac{(\widehat{x}_{k,l} - \mu_{k,l})^2}{\sigma^2_{k,l}} \right] \right)^{-1}, \qquad (4)$$

where $\widehat{x}_{k,l}$ denote the mean estimates and $\mu_{k,l}$ and $\sigma^2_{k,l}$ denote the true mean and variance of $x_{k,l} \mid y_{1:k}$ obtained from the Kalman filter. The expectation in (4) is approximated by averaging over 100 independent runs of the involved algorithms. The ESS reflects the estimator accuracy, obvious by the definition which is tightly related to the mean-squared-error. Intuitively the ESS corresponds to the equivalent number of i.i.d. samples needed for the same accuracy.

We use $N = 500$ and $M = 2 \cdot d$ for NSMC and match the computational time for ST-PF and bootstrap PF. We report the results in Figure 2. Note that the bootstrap PF is omitted from $d = 100, 200$ due to its poor performance already for $d = 50$ (which is to be expected). Each dimension $l = 1, \ldots, d$ provides us with a value of the ESS, so we present the median (lines) and 15–85% percentiles (shaded regions) in the first row of Figure 2.

We have conducted additional experiments with different model parameters and different choices for N and M (some additional results are given in Naesseth et al. (2015)). Overall the results seem to be in agreement with the ones presented here, however ST-PF seems to be more robust to the trade-off between N and M. A rule-of-thumb for NSMC is to generally try to keep N as high as possible, while still maintaining a reasonable estimate of Z_{q_k}.

6.2. Non-Gaussian State Space Model

Next, we consider an example with a non-Gaussian SSM, borrowed from Beskos et al. (2014a) where the full details of the model are given. The transition probability $p(x_k \mid x_{k-1})$ is a localised Gaussian mixture and the measurement probability $p(y_k \mid x_k)$ is t-distributed. The model dimension is $d = 1\,024$. Beskos et al. (2014a) report improvements for ST-PF over both the bootstrap PF and the block PF by Rebeschini & van Handel (2015). We use $N = M = 100$ for both ST-PF and NSMC (the special structure of this model implies that there is no significant computational overhead from running backward sampling) and the bootstrap PF is given $N = 10\,000$. In Figure 3 we report the ESS (4), estimated according to Carpenter et al. (1999). The ESS for the bootstrap PF is close to 0, for ST-PF around 1–2, and for NSMC slightly higher at 7–8. However, we note that all methods perform quite poorly on this model, and to obtain satisfactory results it would be necessary to use more particles.

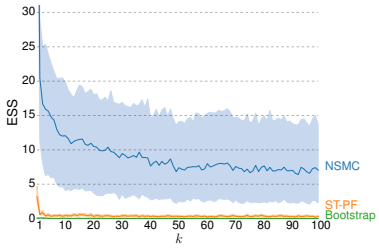

Figure 3. Median ESS with $15 - 85\%$ percentiles (shaded region) for the non-Gaussian SSM.

6.3. Spatio-Temporal Model – Drought Detection

In this final example we study the problem of detecting droughts based on measured precipitation data (Jones & Harris, 2013) for different locations on earth. We look at the situation in North America during the years 1901–1950 and the Sahel region in Africa during the years 1950–2000, time frames including the so-called Dust Bowl in the US during the 1930s (Schubert et al., 2004) and the decades long drought in the Sahel region in Africa starting in the 1960s (Foley et al., 2003; Hoerling et al., 2006).

We consider the spatio-temporal model defined by Fu et al. (2012) and compare with the results therein. Each location in a region is modelled to be in either a *normal* state

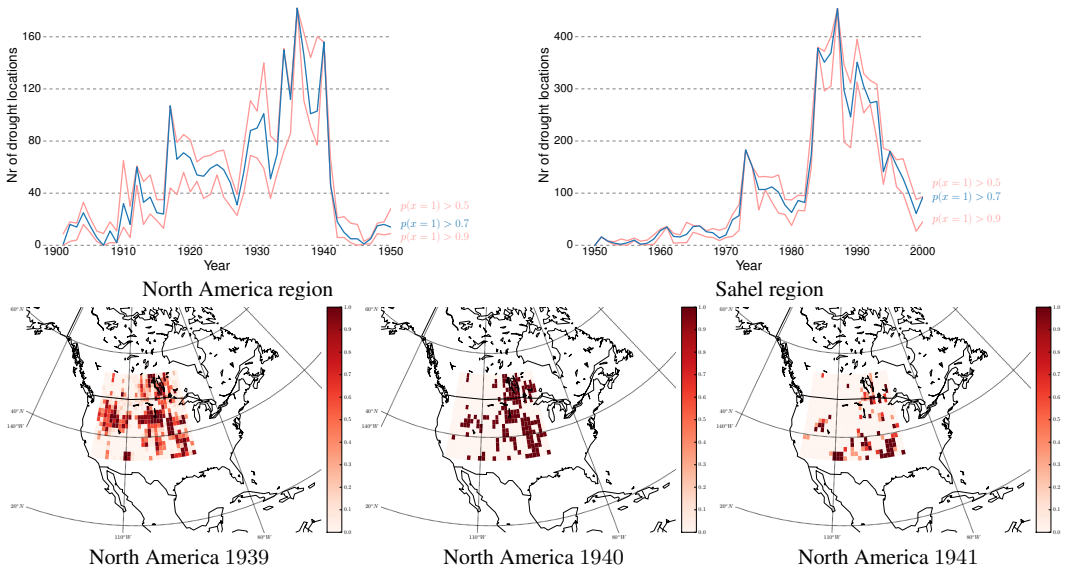

Figure 4. Top: Number of locations with estimated $p(x = 1) > \{0.5, 0.7, 0.9\}$ for the two regions. *Bottom:* Estimate of $p(x_{t,i} = 1)$ for all sites over a span of 3 years. All results for $N = 100$, $N_1 = \{30, 40\}$, $N_2 = 20$.

0 or in an *abnormal* state 1 (drought). Measurements are given by precipitation (in millimeters) for each location and year. At every time instance k our latent structure is described by a rectangular 2D grid $X_k = \{X_{k,i,j}\}_{i=1,j=1}^{I,J}$; in essence this is the model showcased in Figure 1. Fu et al. (2012) considers the problem of finding the maximum aposteriori configuration, using a linear programming relaxation. We will instead compute an approximation of the full posterior filtering distribution $\bar{\pi}_k(x_k) = p(x_k \mid y_{1:k})$. The rectangular structure is used to instantiate an NSMC method that on the first level targets the full posterior filtering distribution, second level the columns, and third level the rows of X_k. Properly weighted samples are generated using a bootstrap PF for the third level. The structure of our NSMC method applied to this particular problem is illustrated in Figure 5.

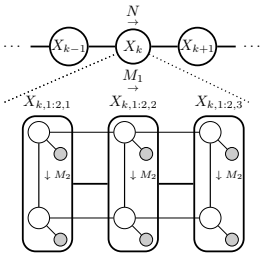

Figure 5. Illustration of the three-level NSMC.

Figure 4 gives the results on the parts of North America that we consider. The first row shows the number of locations where the estimate of $p(x_{k,i,j} = 1)$ exceeds $\{0.5, 0.7, 0.9\}$, for both regions. These results seems to be in agreement with Fu et al. (2012, Figures 3, 6). However, we also receive an approximation of the full posterior and can visualise uncertainty in our estimates, as illustrated by the three different levels of posterior probability for drought. In general, we obtain a rich sample diversity from the posterior distribution. However, for some problematic years the sampler degenerates, with the result that the three credibility levels all coincide. This is also visible in the second row of Figure 4, where we show the posterior estimates $p(x_{k,i,j} \mid y_{1:k})$ for the years 1939–1941, overlayed on the regions of interest. Naturally, one way to improve the estimates is to run the sampler with a larger number of particles, which has been kept very low in this proof-of-concept.

We have shown that a straightforward NSMC implementation with fairly few particles can attain reasonable approximations to the filtering problem for dimensions in the order of hundreds, or even thousands. This means that NSMC methods takes the SMC framework an important step closer to being viable for high-dimensional statistical inference problems. However, NSMC is not a silver bullet for solving high-dimensional inference problems, and the approximation accuracy will be highly model dependent. Hence, much work remains to be done, for instance on combining NSMC with other techniques for high-dimensional inference such as localisation (Rebeschini & van Handel, 2015) and annealing (Beskos et al., 2014b), in order to solve even more challenging problems.

Acknowledgments

This work was supported by the projects: *Learning of complex dynamical systems* (Contract number: 637-2014-466) and *Probabilistic modeling of dynamical systems* (Contract number: 621-2013-5524), both funded by the Swedish Research Council.

References

Andrieu, C. and Roberts, G. O. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

Andrieu, Christophe, Doucet, Arnaud, and Holenstein, Roman. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

Beskos, A., Crisan, D., Jasra, A., Kamatani, K., and Zhou, Y. A stable particle filter in high-dimensions. *ArXiv:1412.3501*, December 2014a.

Beskos, Alexandros, Crisan, Dan, and Jasra, Ajay. On the stability of sequential Monte Carlo methods in high dimensions. *Ann. Appl. Probab.*, 24(4):1396–1445, 08 2014b.

Bickel, Peter, Li, Bo, and Bengtsson, Thomas. *Sharp failure rates for the bootstrap particle filter in high dimensions*, volume Volume 3 of *Collections*, pp. 318–329. Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2008.

Briggs, Jonathan, Dowd, Michael, and Meyer, Renate. Data assimilation for large-scale spatio-temporal systems using a location particle smoother. *Environmetrics*, 24(2):81–97, 2013.

Cappé, Olivier, Moulines, Eric, and Rydén, Tobias. *Inference in Hidden Markov Models*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387402640.

Carpenter, J., Clifford, P., and Fearnhead, P. Improved particle filter for nonlinear problems. *IEE Proceedings Radar, Sonar and Navigation*, 146(1):2–7, 1999.

Chen, Tianshi, Schön, Thomas B., Ohlsson, Henrik, and Ljung, Lennart. Decentralized particle filter with arbitrary state decomposition. *IEEE Transactions on Signal Processing*, 59(2):465–478, Feb 2011.

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. SMC2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426, 2013.

Cohen, Jacques. Bioinformaticsan introduction for computer scientists. *ACM Computing Surveys (CSUR)*, 36 (2):122–158, 2004.

Cressie, N. and Wikle, C. K. *Statistics for spatio-temporal data*. Wiley, 2011.

Crisan, D. and Míguez, J. Nested particle filters for online parameter estimation in discrete-time state-space Markov models. *ArXiv:1308.1883*, August 2013.

Del Moral, P. *Feynman-Kac Formulae - Genealogical and Interacting Particle Systems with Applications*. Probability and its Applications. Springer, 2004.

Djuric, Petar M and Bugallo, Mónica F. Particle filtering for high-dimensional systems. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*, pp. 352–355. IEEE, 2013.

Doucet, A. and Johansen, A. M. A tutorial on particle filtering and smoothing: Fifteen years later. In Crisan, D. and Rozovsky, B. (eds.), *Nonlinear Filtering Handbook*. Oxford University Press, 2011.

Doucet, Arnaud, De Freitas, Nando, and Gordon, Neil. *An introduction to sequential Monte Carlo methods*. Springer, 2001.

Fearnhead, Paul, Papaspiliopoulos, Omiros, Roberts, Gareth O., and Stuart, Andrew. Random-weight particle filtering of continuous time processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):497–512, 2010a.

Fearnhead, Paul, Wyncoll, David, and Tawn, Jonathan. A sequential smoothing algorithm with linear computational cost. *Biometrika*, 97(2):447–464, 2010b.

Foley, J. A., Coe, M. T., Scheffer, M., and Wang, G. Regime shifts in the sahara and sahel: Interactions between ecological and climatic systems in northern africa. *Ecosystems*, 6:524–539, 2003.

Fu, Qiang, Banerjee, Arindam, Liess, Stefan, and Snyder, Peter K. Drought detection of the last century: An MRF-based approach. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pp. 24–34, Anaheim, CA, USA, April 2012.

Godsill, S. J., Doucet, A., and West, M. Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168, March 2004.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107 –113, April 1993.

Hoerling, M., Hurrell, J., Eischeid, J., and Phillips, A. Detection and attribution of twentieth-century northern and southern african rainfall change. *Journal of Climate*, 19: 3989–4008, 2006.

Johansen, A. M., Whiteley, N., and Doucet, A. Exact approximation of Rao-Blackwellised particle filters. In *Proceesings of the 16th IFAC Symposium on System Identification (SYSID)*, pp. 488–493, Brussels, Belgium, 2012.

Jones, P.D. and Harris, I. CRU TS3.21: Climatic research unit (CRU) time-series (ts) version 3.21 of high resolution gridded data of month-by-month variation in climate (jan. 1901- dec. 2012). NCAS British Atmospheric Data Centre, sep 2013. URL `http://dx.doi.org/10.5285/D0E1585D-3417-485F-87AE-4FCECF10A992`.

Kalman, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45, 1960.

Lindsten, F. and Schön, T. B. Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning*, 6(1):1–143, 2013.

Liu, Jun S. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2001.

Monteleoni, Claire, Schmidt, Gavin A., Alexander, Francis, Niculescu-Mizil, Alexandru, Steinhaeuser, Karsten, Tippett, Michael, Banerjee, Arindam, Blumenthal, M. Benno, Auroop R. Ganguly, Jason E. Smerdon, and Tedesco, Marco. Climate informatics. In Yu, Ting, Chawla, Nitesh, and Simoff, Simeon (eds.), *Computational Intelligent Data Analysis for Sustainable Development*. Chapman and Hall/CRC, London, 2013.

Naesseth, Christian A., Lindsten, Fredrik, and Schön, Thomas B. Capacity estimation of two-dimensional channels using sequential Monte Carlo. In *The 2014 IEEE Information Theory Workshop (ITW)*, pp. 431–435, Nov 2014a.

Naesseth, Christian A, Lindsten, Fredrik, and Schön, Thomas B. Sequential Monte Carlo for graphical models. In *Advances in Neural Information Processing Systems 27*, pp. 1862–1870. Curran Associates, Inc., 2014b.

Naesseth, Christian A., Lindsten, Fredrik, and Schön, Thomas B. Nested sequential Monte Carlo methods. *arXiv:1502.02536*, 2015.

Pitt, Michael K and Shephard, Neil. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

Rebeschini, P. and van Handel, R. Can local particle filters beat the curse of dimensionality? *Ann. Appl. Probab. (to appear)*, 2015.

Rue, H. and Held, L. *Gaussian Markov Random Fields, Theory and Applications*. CDC Press, Boca Raton, FL, USA, 2005.

Schubert, S. D., Suarez, M. J., Pegion, P. J., Koster, R. D., and Bacmeister, J. T. On the cause of the 1930s dust bowl. *Science*, 303:1855–1859, 2004.

Shumway, R. H. and Stoffer, D. S. *Time Series Analysis and Its Applications – with R examples*. Springer Texts in Statistics. Springer, New York, USA, third edition, 2011.

Vergé, Christelle, Dubarry, Cyrille, Del Moral, Pierre, and Moulines, Eric. On parallel implementation of sequential Monte Carlo methods: the island particle model. *Statistics and Computing*, pp. 1–18, 2013.

Wainwright, Martin J and Jordan, Michael I. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2): 1–305, 2008.

Wikle, C. K. Modern perspectives on statistics for spatio-temporal data. *WIREs Computational Statistics*, 7(1): 86–98, 2015.