Elements of Sequential Monte Carlo

Other titles in Foundations and Trends® in Machine Learning

Computational Optimal Transport
Gabriel Peyre and Marco Cuturi
ISBN: 978-1-68083-550-2

An Introduction to Deep Reinforcement Learning
Vincent Francois-Lavet, Peter Henderson, Riashat Islam,
Marc G. Bellemare and Joelle Pineau
ISBN: 978-1-68083-538-0

An Introduction to Wishart Matrix Moments
Adrian N. Bishop, Pierre Del Moral and Angele Niclas
ISBN: 978-1-68083-506-9

A Tutorial on Thompson Sampling
Daniel J. Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband
and Zheng Wen
ISBN: 978-1-68083-470-3

Elements of Sequential Monte Carlo

Christian A. Naesseth
Columbia University
christian.a.naesseth@columbia.edu

Fredrik Lindsten
Linköping University
fredrik.lindsten@liu.se

Thomas B. Schön
Uppsala University
thomas.schon@it.uu.se

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Machine Learning

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

C. A. Naesseth, F. Lindsten, T. B. Schön. *Elements of Sequential Monte Carlo*. Foundations and Trends® in Machine Learning, vol. 12, no. 3, pp. 187–306, 2019.

ISBN: 978-1-68083-633-2

Foundations and Trends® in Machine Learning
Volume 12, Issue 3, 2019

Editorial Scope

Topics

Foundations and Trends® in Machine Learning publishes survey and tutorial articles in the following topics:

- Adaptive control and signal processing
- Applications and case studies
- Behavioral, cognitive and neural learning
- Bayesian learning
- Classification and prediction
- Clustering
- Data mining
- Dimensionality reduction
- Evaluation
- Game theoretic learning
- Graphical models
- Independent component analysis
- Inductive logic programming
- Kernel methods
- Markov chain Monte Carlo
- Model choice
- Nonparametric methods
- Online learning
- Optimization
- Reinforcement learning
- Relational learning
- Robustness
- Spectral methods
- Statistical learning theory
- Variational inference
- Visualization

Information for Librarians

Foundations and Trends® in Machine Learning, 2019, Volume 12, 6 issues. ISSN paper version 1935-8237. ISSN online version 1935-8245. Also available as a combined paper and online subscription.

Contents

Elements of Sequential Monte Carlo

Christian A. Naesseth[1], Fredrik Lindsten[2] and Thomas B. Schön[3]

[1]*Columbia University; christian.a.naesseth@columbia.edu*
[2]*Linköping University; fredrik.lindsten@liu.se*
[3]*Uppsala University; thomas.schon@it.uu.se*

ABSTRACT

A core problem in statistics and probabilistic machine learning is to compute probability distributions and expectations. This is the fundamental problem of Bayesian statistics and machine learning, which frames all inference as expectations with respect to the posterior distribution. The key challenge is to approximate these intractable expectations. In this tutorial, we review sequential Monte Carlo (SMC), a random-sampling-based class of methods for approximate inference. First, we explain the basics of SMC, discuss practical issues, and review theoretical results. We then examine two of the main user design choices: the *proposal distributions* and the so called *intermediate target distributions*. We review recent results on how variational inference and amortization can be used to learn efficient proposals and target distributions. Next, we discuss the SMC estimate of the normalizing constant, how this can be used for pseudo-marginal inference and inference evaluation. Throughout the tutorial we illustrate the use of SMC on various models commonly used in machine learning, such as stochastic recurrent neural networks, probabilistic graphical models, and probabilistic programs.

Christian A. Naesseth, Fredrik Lindsten and Thomas B. Schön (2019), Elements of Sequential Monte Carlo, Foundations and Trends® in Machine Learning: Vol. 12, No. 3, pp 187–306.
DOI: 10.1561/2200000074.

1

Introduction

A key strategy in machine learning is to break down a problem into smaller and more manageable parts, then process data or unknown variables recursively. Well known examples of this are message passing algorithms for graphical models and annealing for optimization or sampling. Sequential Monte Carlo (SMC) is a class of methods that are tailored to solved statistical inference problems recursively. These methods have mostly received attention in the signal processing and statistics communities. With well over two decades of research in SMC, they have enabled inference in increasingly complex and challenging models. Recently, there has been an emergent interest in this class of algorithms from the machine learning community. We have seen applications to probabilistic programming (Wood *et al.*, 2014), variational inference (VI) (Maddison *et al.*, 2017; Naesseth *et al.*, 2018; Le *et al.*, 2018), inference evaluation (Grosse *et al.*, 2015; Cusumano-Towner and Mansinghka, 2017), probabilistic graphical models (PGMs) (Ihler and McAllester, 2009; Naesseth *et al.*, 2014; Paige and Wood, 2016), Bayesian nonparametrics (Fearnhead, 2004) and many other areas.

We provide a unifying view of the SMC methods that have been developed since their conception in the early 1990s (Gordon *et al.*, 1993; Stewart and McCarty, 1992; Kitagawa, 1993). In this introduction we provide relevant background material, introduce a running example, and discuss the use of code snippets throughout the tutorial.

1.1 Historical Background

SMC methods are generic tools for performing approximate (statistical) inference, predominantly Bayesian inference. They use a weighted sample set to iteratively approximate the posterior distribution of a probabilistic model. Ever since the dawn of Monte Carlo methods (see e.g. Metropolis and Ulam (1949) for an early discussion), random sample-based approximations have been recognized as powerful tools for inference in complex probabilistic models. Parallel to the development of Markov chain Monte Carlo (MCMC) methods (Metropolis *et al.*, 1953; Hastings, 1970), sequential importance sampling (SIS) (Handschin and Mayne, 1969) and sampling/importance resampling (Rubin, 1987) laid the foundations for what would one day become SMC.

SMC methods where initially known as particle filters (Gordon *et al.*, 1993; Stewart and McCarty, 1992; Kitagawa, 1993). Particle filters where conceived as algorithms for online inference in nonlinear state space models (SSMs) (Cappé *et al.*, 2005). Since then there has been a flurry of work applying SMC and particle filters to perform approximate inference in ever more complex models. While research in SMC initially focused on SSMs, we will see that SMC can be a powerful tool in a much broader setting.

1.2 Probabilistic Models and Target Distributions

As mentioned above, SMC methods were originally developed as an approximate solution to the so called filtering problem, which amounts to online inference in dynamical models. Several overview and tutorial articles focus on particle filters, i.e. the SMC algorithms specifically tailored to solve the online filtering problem (Arulampalam *et al.*, 2002; Doucet and Johansen, 2009; Fearnhead and Künsch, 2018). In this tutorial we will take a different view and explain how SMC can be used to solve more general “offline” problems. We shall see how this viewpoint opens up for many interesting applications of SMC in machine learning that do not fall in the traditional filtering setup, and furthermore how it gives rise to new and interesting design choices. For complementary review and tutorial articles that treat a similar topic see also Del Moral *et al.* (2006a) and Doucet and Lee (2018).

We consider a generic probabilistic model given by a joint probability distribution function (PDF) of latent variables $\mathbf{x}$ and observed data $\mathbf{y}$,

$$p(\mathbf{x}, \mathbf{y}). \tag{1.1}$$

We focus on Bayesian inference, where the key object is the posterior distribution

$$p(\mathbf{x} \,|\, \mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})}, \tag{1.2}$$

where $p(\mathbf{y})$ is known as the marginal likelihood.

The *target distributions* are a sequence of probability distributions that we recursively approximate using SMC. We define each target distribution $\gamma_t(x_{1:t})$ in the sequence as a joint PDF of latent variables $x_{1:t} = (x_1, \dots, x_t)$, where $t = 1, \dots, T$. The PDF is denoted by

$$\gamma_t(x_{1:t}) := \frac{1}{Z_t} \widetilde{\gamma}_t(x_{1:t}), \quad t = 1, \dots, T, \tag{1.3}$$

where $\widetilde{\gamma}_t$ is a positive integrable function and Z_t is the normalization constant, ensuring that γ_t is indeed a PDF.

We connect the target distributions to the probabilistic model through a requirement on the final target distribution $\gamma_T(x_{1:T})$. We enforce the condition that $\gamma_T(x_{1:T})$ is either equivalent to the posterior distribution, or that it contains the posterior distribution as a marginal distribution. The *intermediate* target distributions, i.e. $\gamma_t(x_{1:t})$ for $t < T$, are useful only insofar they help us approximate the final target $\gamma_T(x_{1:T})$. This approach is distinct from most previous tutorials on particle filters and SMC that traditionally focus on the intermediate targets, i.e. the filtering distributions. We stress that it is not necessarily the case that $x_{1:T}$ (the latent variables of the target distribution) is equal to $\mathbf{x}$ (the latent variables of the probabilistic model of interest), as the former can include additional auxiliary variables.

Below we introduce a few examples of probabilistic models and some straightforward choices of target distributions. We introduce and illustrate our running example which will be used throughout. We will return to the issue of choosing the sequence of intermediate targets in Section 3.2.

State Space Models The state space model (or hidden Markov model) is a type of probabilistic models where the latent variables and data satisfy a Markov property. For this model we typically have $\mathbf{x} = x_{1:T}$. Often the measured data can also be split into a sequence of the same length (T) as the latent variables, i.e. $\mathbf{y} = y_{1:T}$. The model is defined by a transition PDF f and an observation PDF g,

$$x_t \mid x_{t-1} \sim f(\cdot \mid x_{t-1}), \tag{1.4a}$$

$$y_t \mid x_t \sim g(\cdot \mid x_t). \tag{1.4b}$$

The joint PDF is

$$p(\mathbf{x}, \mathbf{y}) = p(x_1) g(y_1 \mid x_1) \prod_{t=2}^{T} f(x_t \mid x_{t-1}) g(y_t \mid x_t), \tag{1.5}$$

where $p(x_1)$ is the prior on the initial state x_1. This class of models is especially common for data that has an inherent temporal structure such as in the field of signal processing. A common choice is to let the target distributions follow the same sequential structure as in Eq. (1.5):

$$\widetilde{\gamma}_t(x_{1:t}) = p(x_1) g(y_1 \mid x_1) \prod_{k=2}^{t} f(x_k \mid x_{k-1}) g(y_k \mid x_k), \tag{1.6}$$

which means that the final normalized target distribution satisfies $\gamma_T(x_{1:T}) = p(\mathbf{x} \mid \mathbf{y})$ as required. This is the model class and target distributions which are studied in the classical filtering setup.

Non-Markovian Latent Variable Models The non-Markovian latent variable models (LVMs) are characterized by either no, or higher order, Markov structure between the latent variables $\mathbf{x}$ and/or data $\mathbf{y}$. This can be seen as a non-trivial extension of the SSM, see Eq. (1.4), which has a Markov structure. Also for this class of models it is common to have $\mathbf{x} = x_{1:T}$ and $\mathbf{y} = y_{1:T}$.

Unlike the SSM, the non-Markovian LVM in its most general setting requires access to all previous latent variables $x_{1:t-1}$ to generate x_t, y_t

$$x_t \mid x_{1:t-1} \sim f_t(\cdot \mid x_{1:t-1}), \tag{1.7a}$$

$$y_t \mid x_{1:t} \sim g_t(\cdot \mid x_{1:t}), \tag{1.7b}$$

where we again refer to f_t and g_t as the transition PDF and observation PDF, respectively. The joint PDF is given by

$$p(\mathbf{x},\mathbf{y}) = p(x_1)g(y_1 \mid x_1)\prod_{t=2}^{T} f_t(x_t \mid x_{1:t-1})g_t(y_t \mid x_{1:t}), \tag{1.8}$$

where $p(x_1)$ is again the prior on x_1. A typical target distribution is given by

$$\widetilde{\gamma}_t(x_{1:t}) = \widetilde{\gamma}_{t-1}(x_{1:t-1})f_t(x_t \mid x_{1:t-1})g_t(y_t \mid x_{1:t}), \quad t > 1, \tag{1.9}$$

with $\widetilde{\gamma}_1(x_1) = p(x_1)g_1(y_1 \mid x_1)$. Another option is

$$\begin{aligned}
\widetilde{\gamma}_1(x_1) &= p(x_1),\\
\widetilde{\gamma}_t(x_{1:t}) &= \widetilde{\gamma}_{t-1}(x_{1:t-1})f_t(x_t \mid x_{1:t-1}), \quad 1 < t < T,\\
\widetilde{\gamma}_T(x_{1:T}) &= \widetilde{\gamma}_{T-1}(x_{1:T-1})f_T(x_T \mid x_{1:T-1})\prod_{t=1}^{T} g_t(y_t \mid x_{1:t}).
\end{aligned}$$

For both these sequences of target distributions the final iteration T is the posterior distribution, i.e. $\gamma_T(x_{1:T}) = p(x_{1:T} \mid y_{1:T}) = p(\mathbf{x} \mid \mathbf{y})$. However, the former one will often lead to more accurate inferences. This is because we introduce information from the data at an earlier stage in the SMC algorithm.

Throughout the monograph we will exemplify the different methods using a Gaussian special case of Eq. (1.7), see Example 1.1. We let the prior on $x_{1:t}$, defined by the transition PDFs $f_1, \ldots f_t$, be Markovian and introduce the non-Markov property instead through the observation PDFs $g_1, \ldots, g_t$.

Example 1.1 (Non-Markovian Gaussian Sequence Model). As our running example for illustration purposes we use a non-Markovian Gaussian sequence model. It is

$$x_t \mid x_{1:t-1} \sim f_t(\cdot \mid x_{t-1}), \qquad y_t \mid x_{1:t} \sim g_t(\cdot \mid x_{1:t}), \tag{1.10}$$

with observed variables y_t (data), and where

$$\begin{aligned}
f_t(x_t \mid x_{t-1}) &= \mathcal{N}(x_t \mid \phi x_{t-1}, q),\\
g_t(y_t \mid x_{1:t}) &= \mathcal{N}\left(y_t \mid \sum_{k=1}^{t} \beta^{t-k} x_k, r\right),
\end{aligned}$$

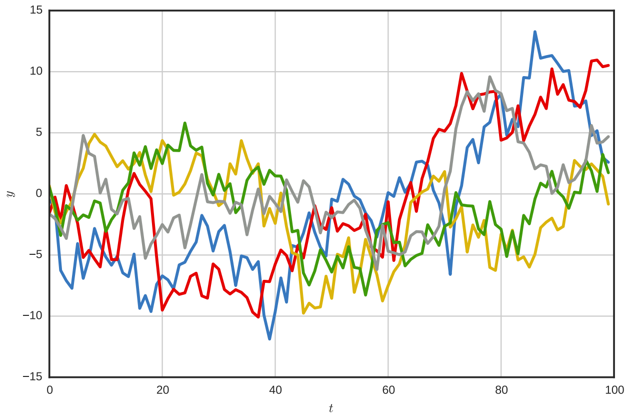

Figure 1.1: Five sample paths of $y_{1:T}$ from our running example for $T = 100$.

where $\mathcal{N}(x \mid \mu, \Sigma)$ denotes a Gaussian distribution on x with mean μ and (co)variance Σ. We let the prior at $t = 1$ be $p(x_1) = \mathcal{N}(x_1 \mid 0, q)$. Artificial data was generated using $(\phi, q, \beta, r) = (0.9, 1, 0.5, 1)$. The distribution of interest is the posterior distribution $p(x_{1:T} \mid y_{1:T})$. We illustrate a few sample paths of $y_{1:T}$ in Fig. 1.1 for $T = 100$.

We can adjust the strength of the dependence on previous latent variables in the observations, y_t, through the parameter $\beta \in [0, 1]$. If we set $\beta = 0$ we obtain a linear Gaussian SSM, since the data depends only on the most recent latent x_t. On the other hand if we let $\beta = 1$, this signifies that x_k for $k < t$ has equally strong effect on y_t as does x_t.

Another example of a non-Markovian LVM often encountered in machine learning is the stochastic recurrent neural network (RNN) (Chung *et al.*, 2015; Bayer and Osendorfer, 2014; Fraccaro *et al.*, 2016; Maddison *et al.*, 2017; Naesseth *et al.*, 2018). We define the stochastic RNN below and will return to it again in Chapter 3.

Example 1.2 (Stochastic Recurrent Neural Network). A stochastic RNN is a non-Markovian LVM where the parameters of the transition and observation models are defined by RNNs. A common example is using the conditional Gaussian distribution to define the transition PDF

$$f_t(x_t \mid x_{1:t-1}) = \mathcal{N}(x_t \mid \mu_t(x_{1:t-1}), \Sigma_t(x_{1:t-1})),$$

where the functions $\mu_t(\cdot), \Sigma_t(\cdot)$ are defined by RNNs.

Conditionally independent models A common model in probabilistic machine learning is to assume that the datapoints y_k in the dataset $\mathbf{y} = \{y_k\}_{k=1}^K$ are conditionally independent given the latent $\mathbf{x}$. This means that the joint PDF is given by

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}) \underbrace{\prod_{k=1}^{K} g_k(y_k \,|\, \mathbf{x})}_{p(\mathbf{y}\,|\,\mathbf{x})}, \tag{1.11}$$

where $p(\mathbf{y}\,|\,\mathbf{x})$ is the likelihood. For this class of models it might not be immediately apparent that we can define a useful sequence of target distributions on latent variables $x_{1:t}$. There is no obvious sequence structure as in the SSM or the non-Markovian LVM. However, as we shall see, we can make use of auxiliary variables to design target distributions that can help with inference.

We will discuss two approaches to design the sequence of target distributions: using data tempering and likelihood tempering, respectively. Both of these will make use of an auxiliary variable technique, where each x_t is a random variable on the same space as $\mathbf{x}$.

Data tempering: Using data tempering we add the data y_k to the target distribution one by one. In this case the data index k coincides with the target index t. We define the target distribution

$$\widetilde{\gamma}_t(x_{1:t}) = p(x_t) \prod_{k=1}^{t} g_k(y_k \,|\, x_t) \cdot \prod_{k=1}^{t-1} s_k(x_k \,|\, x_{k+1}), \tag{1.12}$$

where the distributions $s_k(x_k \,|\, x_{k+1})$ are a design choice, known as *backward kernels* (Chopin, 2002; Del Moral *et al.*, 2006b). With this choice, we have that the marginal distribution of x_T at the final iteration is exactly the posterior distribution, i.e. $\gamma_T(x_T) = p(\mathbf{x}\,|\,\mathbf{y})$. In fact, at each step we have that the marginal target distribution is a partial posterior $\gamma_t(x_t) = p(\mathbf{x}\,|\,y_{1:t})$.

Likelihood tempering: With likelihood tempering, instead of adding the data one by one, we change the likelihood $p(\mathbf{y}\,|\,\mathbf{x})$ through a sequence of positive variables. We define the target distribution

$$\widetilde{\gamma}_t(x_{1:t}) = p(x_t) p(\mathbf{y}\,|\,x_t)^{\tau_t} \cdot \prod_{k=1}^{t-1} s_k(x_k \,|\, x_{k+1}), \tag{1.13}$$

where $0 = \tau_1 < \ldots < \tau_T = 1$, and again make use of user chosen backward kernels $s_k(x_k \mid x_{k+1})$ (Chopin, 2002; Del Moral *et al.*, 2006b). In this setting all data is considered at each iteration. Since $\tau_T = 1$, we have that the final marginal target distribution is again equal to the posterior $\gamma_T(x_T) = p(\mathbf{x} \mid \mathbf{y})$.

Applying SMC methods to tempered (and similar) target distributions has been studied by e.g. Chopin (2002) and Del Moral *et al.* (2006b). If the proposal $q_{k+1}(x_{k+1} \mid x_k)$ is a Markov kernel with stationary distribution $\gamma_{k+1}(x_{k+1})$, then a commonly used backward kernel is $s_k(x_k \mid x_{k+1}) = \frac{\gamma_{k+1}(x_k) q_{k+1}(x_{k+1} \mid x_k)}{\gamma_{k+1}(x_{k+1})}$. We refer to the works of Chopin (2002) and Del Moral *et al.* (2006b) for a thorough discussion on the choice of backward kernels $s_k(x_k \mid x_{k+1})$. Another well known example is *annealed importance sampling* by Neal (2001), which uses the backward kernel example above to define the target distributions, but relies on SIS instead of SMC for inference.

Models and Targets We have seen several probabilistic models with examples of corresponding target distributions. While not limited to these, this illustrates the wide range of the applicability of SMC. In fact, as long as we can design a sequence of target distributions such that γ_T coincides with the distribution of interest, we can leverage SMC for inference.

1.3 Applications

Sequential Monte Carlo and importance sampling methods have already seen a plethora of applications to machine learning and statistical inference problems. Before we turn to the fundamentals of the various algorithms it can be helpful to understand some of these applications. We present and discuss a few select examples of applications of SMC and importance sampling (IS) to probabilistic graphical models, Bayesian nonparametric models, probabilistic programming, and inference evaluation.

Probabilistic Graphical Models Probabilistic graphical models (PGM; see e.g. Koller *et al.* (2009) and Wainwright and Jordan (2008)) are probabilistic models where the conditional independencies in the joint PDF are described by edges in a graph. The graph structure allows for easy and strong control on the type of prior information that the user can express.

The main limitation of the PGM is that exact inference is often intractable and approximate inference is challenging.

The PGM is a probabilistic model where the PDF factorizes according to an underlying graph described by a set of cliques $C \in \mathscr{C}$, i.e. fully connected subsets of the vertices $V \in \mathscr{V}$ where $\mathscr{V}$ contains all individual components of $\mathbf{x}$ and $\mathbf{y}$. The undirected graphical model can be denoted by

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{C \in \mathscr{C}} \psi_C(x_C), \tag{1.14}$$

where x_C includes all elements of $\mathbf{x}$ and $\mathbf{y}$ in the clique C, and Z is a normalization constant ensuring that the right hand side is a proper PDF.

SMC methods have recently been successfully applied to the task of inference in general PGMs, see e.g. MacEachern *et al.* (1999), Chopin (2002), Del Moral *et al.* (2006b), Ihler and McAllester (2009), Naesseth *et al.* (2014), Paige and Wood (2016), Lindsten *et al.* (2017), and Lindsten *et al.* (2018) for representative examples.

Probabilistic Programming Probabilistic programming languages are programming languages designed to describe probabilistic models and to automate the process of doing inference in those models. We can think of probabilistic programming as that of automating statistical inference, particularly Bayesian inference, using tools from computer science and computational statistics. Developing a syntax and semantics to describe and denote probabilistic models and the inference problems we are interested in solving is key to the probabilistic programming language. To define what separates a probabilistic programming language from a standard programming language we quote Gordon *et al.* (2014): “Probabilistic programs are usual functional or imperative programs with two added constructs: (1) the ability to draw values at random from distributions, and (2) the ability to condition values of variables in a program via observations.” This aligns very well with the notion of Bayesian inference through the posterior distribution Eq. (1.2); through the syntax of the language we define $\mathbf{x}$ to be the random values we sample and $\mathbf{y}$ our observations that we condition on through the use of Bayes rule. The probabilistic program then essentially defines our joint probabilistic model $p(\mathbf{x}, \mathbf{y})$.

One of the main challenges of probabilistic programming is to develop algorithms that are general enough to enable inference for any model (probabilistic program) that we could conceivably write down using the language. Recently Wood *et al.* (2014) have shown that SMC-based approaches can be used as inference back-ends in probabilistic programs.

For a more thorough treatment of probabilistic programming we refer the interested reader to the recent tutorial by Meent *et al.* (2018) and the survey by Gordon *et al.* (2014).

Bayesian nonparametric models Nonparametric models are characterized by having a complexity which grows with the amount of available data. In a Bayesian context this implies that the usual latent random variables (i.e., parameters) of the model are replaced by latent stochastic processes. Examples include Gaussian processes, Dirichlet processes, and Beta processes; see e.g. Hjort *et al.* (2010) for a general introduction.

Sampling from these latent stochastic processes, conditionally on observed data, can be done using SMC. To give a concrete example, consider the Dirichlet process mixture model, which is a clustering model that can handle an unknown and conceptually infinite number of mixture components. Let y_t, $t = 1, 2, \ldots$ be a stream of data. Let x_t, $t = 1, 2, \ldots$ be a sequence of latent integer-valued random variables, such that x_t is the index of the mixture component to which datapoint y_t belongs. A generative representation of the mixture assignment variables is given by

$$p(x_{t+1} = j \mid x_{1:t}) = \begin{cases} \frac{n_{t,j}}{t+\alpha} & \text{for } j = 1, \ldots, J_t, \\ \frac{\alpha}{t+\alpha} & \text{for } j = J_t + 1, \end{cases}$$

where $J_t := \max\{x_{1:t}\}$ is the number of distinct mixture components represented by the first t datapoints, and $n_{t,j} := \sum_{k=1}^{t} \mathbb{I}\{x_k = j\}$ is the number of datapoints among $y_{1:t}$ that belong to the jth component.

The model is completed by specifying the distribution of the data, conditionally on the mixture assignment variables:

$$\theta_k \sim F(\theta), \quad k = 1, 2, \ldots$$
$$p(y_t \mid x_t, \{\theta_k\}_{k \geq 1}) = G(y_t \mid \theta_{x_t}),$$

where $G(\cdot \mid \theta)$ is an emission probability distribution parameterized by θ and $F(\theta)$ is a prior distribution over the mixture parameters θ.

Note that the mixture assignment variables x_t, $t = 1, 2, \ldots$ evolve according to a latent stochastic process. Solving the clustering problem amounts to computing the posterior distribution of this stochastic process, conditionally on the observed data. One way to address this problem is to use SMC; see Fearnhead (2004) for an efficient implementation tailored to the discrete nature of the problem.

Inference Evaluation An important problem when performing approximate Bayesian inference is to figure out when our approximation is "good enough"? Is it possible to give practical guarantees on the approximation we obtain? We need ways to evaluate how accurate our approximate inference algorithms are when compared to the true target distribution that we are trying to approximate. We will refer to the process of evaluating and validating approximate inference methods as *inference evaluation*.

Inference evaluation is mainly concerned with measuring how close our approximation is to the true object we are trying to estimate, often a posterior distribution. For simulated data, Grosse *et al.* (2015) and Grosse *et al.* (2016) have shown that we can make use of SMC and IS to bound the symmetrized Kullback-Leibler (KL) divergence between our approximate posterior and the true posterior. In another related work Cusumano-Towner and Mansinghka (2017) have shown that SMC-based methods show promise in estimating the symmetric KL divergence between the approximate posterior and a gold standard algorithm.

1.4 Example Code

We will be making use of inline Python code snippets throughout the manuscript to illustrate the algorithms and methods. Below we summarize the modules that are necessary to import to run the code snippets:

```python
import numpy as np
import numpy.random as npr
from scipy.misc import logsumexp
from scipy.stats import norm
```

Example Code 1.1: Necessary imports for Python code examples.

1.5 Outline

The remainder of this tutorial is organized as follows. In Chapter 2, we first introduce IS, a foundational building block for SMC. Then, we discuss the limitations of IS and how SMC resolves these. Finally, the section concludes with discussing some practical issues and theoretical results relevant to SMC methods.

Chapter 3 is focused on the two key design choices of SMC: the *proposal* and *target* distributions. Initially we focus on the proposal, discussing various ways of adapting and learning good proposals that will make the approximation more accurate. Then we discuss the sequence of target distributions; how we can learn intermediate distributions that help us when we try to approximate the posterior.

Chapter 4 focuses on pseudo marginal (PM) methods and other SMC methods that rely on a concept known as *proper weights*. First, we provide a simple and straightforward proof of the unbiasedness property of the SMC normalization constant estimate. Then, we describe and illustrate the combination of MCMC and SMC methods through PM algorithms. We move on to detail properly weighted SMC, a concept that unites and extends the random weights and nested SMC algorithms. Finally, we conclude the chapter by considering a few approaches for distributed and parallel SMC.

In Chapter 5 we introduce conditional sequential Monte Carlo (CSMC), and related methods for simulating from and computing expectations with respect to a target distribution. First, we introduce the basic CSMC algorithm and provide a straightforward proof of the unbiasedness of the inverse normalization constant estimate. Then, we show how SMC and CSMC can be combined to leverage multi-core and parallel computing architectures in the interacting particle Markov chain Monte Carlo (IPMCMC) algorithm. Finally, we discuss recent applications of CSMC for evaluating approximate inference methods. The tutorial concludes with a discussion and outlook in Chapter 6.

References

Aksan, E. and O. Hilliges (2019). “STCN: Stochastic temporal convolutional networks”. In: *International Conference on Learning Representations (ICLR)*. New Orleans, LA, USA.

Anderson, B. and J. Moore (1979). *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall.

Andrieu, C., A. Doucet, and R. Holenstein (2010). “Particle Markov chain Monte Carlo methods”. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 72(3): 269–342.

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

Arulampalam, M. S., S. Maskell, N. Gordon, and T. Clapp (2002). “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking”. *IEEE Transactions on signal processing*. 50(2): 174–188.

Bai, S., J. Z. Kolter, and V. Koltun (2018). “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling”. *arXiv:1803.01271*.

Bayer, J. and C. Osendorfer (2014). “Learning Stochastic Recurrent Networks”. *arXiv:1411.7610*.

Bengtsson, T., P. Bickel, and B. Li (2008). "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems". In: *Probability and Statistics: Essays in Honor of David A. Freedman*. Ed. by D. Nolan and T. Speed. Vol. Volume 2. *Collections*. Institute of Mathematical Statistics. 316–334.

Beskos, A., D. Crisan, and A. Jasra (2014). "On the stability of sequential Monte Carlo methods in high dimensions". *The Annals of Applied Probability*. 24(4): 1396–1445.

Beskos, A., O. Papaspiliopoulos, G. O. Roberts, and P. Fearnhead (2006). "Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion)". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 68(3): 333–382.

Bickel, P., B. Li, and T. Bengtsson (2008). "Sharp failure rates for the bootstrap particle filter in high dimensions". In: *Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*. Ed. by B. Clarke and S. Ghosal. Vol. Volume 3. *Collections*. Institute of Mathematical Statistics. 318–329.

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2017). "Variational inference: A review for statisticians". *Journal of the American Statistical Association*. 112(518): 859–877.

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

Buchholz, A., N. Chopin, and P. E. Jacob (2018). "Adaptive Tuning Of Hamiltonian Monte Carlo Within Sequential Monte Carlo". *arXiv:1808.07730*.

Cappé, O., E. Moulines, and T. Rydén (2005). *Inference in Hidden Markov Models*. Springer-Verlag New York.

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

Chopin, N. (2002). "A sequential particle filter method for static models". *Biometrika*. 89(3): 539–552.

Chopin, N. (2004). "Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference". *The Annals of Statistics*. 32: 2385–2411.

Chopin, N., P. E. Jacob, and O. Papaspiliopoulos (2013). "SMC^2: an efficient algorithm for sequential analysis of state-space models". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 75(3): 397–426.

Chung, J., K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio (2015). "A Recurrent Latent Variable Model for Sequential Data". In: *Advances in Neural Information Processing Systems*.

Cornebise, J., É. Moulines, and J. Olsson (2008). "Adaptive methods for sequential importance sampling with application to state space models". *Statistics and Computing*. 18(4): 461–480.

Cusumano-Towner, M. F. and V. K. Mansinghka (2017). "AIDE: An algorithm for measuring the accuracy of probabilistic inference algorithms". In: *Advances in Neural Information Processing Systems*. 3004–3014.

Dahlin, J., F. Lindsten, J. Kronander, and T. B. Schön (2015). "Accelerating pseudo-marginal Metropolis-Hastings by correlating auxiliary variables". *arXiv:1511.05483*.

Del Moral, P., A. Doucet, and A. Jasra (2006a). "Sequential Monte Carlo methods for Bayesian Computation". *Bayesian Statistics 8*.

Del Moral, P. (2004). *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer Verlag.

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

Del Moral, P. and A. Guionnet (2001). "On the stability of interacting processes with applications to filtering and genetic algorithms". In: *Annales de l'IHP Probabilités et statistiques*. Vol. 37. No. 2. 155–194.

Deligiannidis, G., A. Doucet, and M. K. Pitt (2018). "The correlated pseudo-marginal method". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 80(5): 839–870.

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

Douc, R. and O. Cappé (2005). "Comparison of resampling schemes for particle filtering". In: *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*. IEEE. 64–69.

Doucet, A. and A. Lee (2018). “Sequential Monte Carlo methods”. *Handbook of Graphical Models*: 165–189. Ed. by M. Maathuis, M. Drton, S. L. Lauritzen, and M. Wainwrigth.

Doucet, A., M. Briers, and S. Sénécal (2006). “Efficient block sampling strategies for sequential Monte Carlo methods”. *Journal of Computational and Graphical Statistics*. 15(3): 693–711.

Doucet, A., S. Godsill, and C. Andrieu (2000). “On sequential Monte Carlo sampling methods for Bayesian filtering”. *Statistics and computing*. 10(3): 197–208.

Doucet, A. and A. M. Johansen (2009). “A tutorial on particle filtering and smoothing: Fifteen years later”. *Handbook of nonlinear filtering*. 12(656-704): 3.

Fearnhead, P. (2004). “Particle filters for mixture models with an unknown number of components”. *Statistics and Computing*. 14: 11–21.

Fearnhead, P. (1998). “Sequential Monte Carlo methods in filter theory”. *PhD thesis*. University of Oxford Oxford.

Fearnhead, P. and H. R. Künsch (2018). “Particle filters and data assimilation”. *Annual Review of Statistics and Its Application*. 5: 421–449.

Fearnhead, P., O. Papaspiliopoulos, G. O. Roberts, and A. Stuart (2010). “Random-weight particle filtering of continuous time processes”. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 72(4): 497–512.

Fearnhead, P. and B. M. Taylor (2013). “An adaptive sequential Monte Carlo sampler”. *Bayesian analysis*. 8(2): 411–438.

Fraccaro, M., S. K. Sønderby, U. Paquet, and O. Winther (2016). “Sequential neural models with stochastic layers”. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Gerber, M., N. Chopin, and N. Whiteley (2017). “Negative association, ordering and convergence of resampling methods”. *arXiv:1707.01845*.

Gershman, S. and N. Goodman (2014). “Amortized inference in probabilistic reasoning”. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 36. No. 36.

Glynn, P. W. and C.-H. Rhee (2014). “Exact estimation for Markov chain equilibrium expectations”. *Journal of Applied Probability*. 51A: 377–389.

Gordon, A. D., T. A. Henzinger, A. V. Nori, and S. K. Rajamani (2014). "Probabilistic programming". In: *Proceedings of the on Future of Software Engineering*. ACM. 167–181.

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

Grosse, R. B., S. Ancha, and D. M. Roy (2016). "Measuring the reliability of MCMC inference with bidirectional Monte Carlo". In: *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc. 2451–2459.

Grosse, R. B., Z. Ghahramani, and R. P. Adams (2015). "Sandwiching the marginal likelihood using bidirectional Monte Carlo". *arXiv:1511.02543*.

Gu, S., Z. Ghahramani, and R. E. Turner (2015). "Neural adaptive sequential Monte Carlo". In: *Advances in Neural Information Processing Systems*. 2629–2637.

Guarniero, P., A. M. Johansen, and A. Lee (2017). "The iterated auxiliary particle filter". *Journal of the American Statistical Association*. 112(520): 1636–1647.

Gulrajani, I., K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville (2017). "PixelVAE: A latent variable model for natural images". In: *International Conference on Representation Learning*.

Handschin, J. E. and D. Q. Mayne (1969). "Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering". *International Journal of Control*. 9(5): 547–559.

Hastings, W. K. (1970). "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". *Biometrika*. 57(1): 97–109.

Heng, J., A. N. Bishop, G. Deligiannidis, and A. Doucet (2017). "Controlled Sequential Monte Carlo". *arXiv:1708.08396*.

Hjort, N. L., C. Holmes, P. Müller, and S. G. Walker, eds. (2010). *Bayesian Nonparametrics*. Cambridge University Press.

Hol, J. D., T. B. Schon, and F. Gustafsson (2006). "On resampling algorithms for particle filters". In: *IEEE Nonlinear Statistical Signal Processing Workshop*. IEEE. 79–82.

Huggins, J. H. and D. M. Roy (2015). "Sequential Monte Carlo as Approximate Sampling: bounds, adaptive resampling via ∞-ESS, and an application to Particle Gibbs". *arXiv:1503.00966*.

Ihler, A. and D. McAllester (2009). “Particle belief propagation”. In: *Artificial Intelligence and Statistics*. 256–263.

Ionides, E. L., D. Nguyen, Y. Atchadé, S. Stoev, and A. A. King (2015). “Inference for dynamic and latent variable models via iterated, perturbed Bayes maps”. *Proceedings of the National Academy of Sciences*. 112(3): 719–724.

Jacob, P. E., F. Lindsten, and T. B. Schön (2019). “Smoothing with couplings of conditional particle filters”. *Journal of the American Statistical Association (JASA)*.

Jacob, P. E. and A. H. Thiery (Apr. 2015). “On nonnegative unbiased estimators”. *The Annals of Statistics*. 43(2): 769–784.

Jacob, P. E., L. M. Murray, and S. Rubenthaler (2015). “Path storage in the particle filter”. *Statistics and Computing*. 25(2): 487–496.

Jasra, A., D. A. Stephens, A. Doucet, and T. Tsagaris (2011). “Inference for Lévy-driven stochastic volatility models via adaptive sequential Monte Carlo”. *Scandinavian Journal of Statistics*. 38(1): 1–22.

Johansen, A. M. and A. Doucet (2008). “A note on auxiliary particle filters”. *Statistics & Probability Letters*. 78(12): 1498–1504.

Julier, S. J. (2002). “The scaled unscented transformation”. In: *Proceedings of the 2002 American Control Conference*. Vol. 6. IEEE. 4555–4559.

Julier, S. J. and J. K. Uhlmann (1997). “New extension of the Kalman filter to nonlinear systems”. In: *Signal processing, sensor fusion, and target recognition VI*. Vol. 3068. International Society for Optics and Photonics. 182–194.

Kahn, H. (1950a). “Random sampling (Monte Carlo) techniques in neutron attenuation problems–I.” *Nucleonics*. 6(5): 27–37.

Kahn, H. (1950b). “Random sampling (Monte Carlo) techniques in neutron attenuation problems–II.” *Nucleonics*. 6(6): 60–65.

Kingma, D. P. and M. Welling (2014). “Auto-Encoding Variational Bayes”. In: *International Conference on Learning Representations*.

Kitagawa, G. (1993). “A Monte Carlo filtering and smoothing method for non-Gaussian nonlinear state space models”. In: *Proceedings of the 2nd US-Japan joint Seminar on Statistical Time Series Analysis*. 110–131.

Kitagawa, G. (1996). “Monte Carlo filter and smoother for non-Gaussian nonlinear state space models”. *Journal of computational and graphical statistics*. 5(1): 1–25.

Koller, D., N. Friedman, and F. Bach (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Krishnan, R. G., U. Shalit, and D. Sontag (2017). "Structured Inference Networks for Nonlinear State Space Models." In: *AAAI*. 2101–2109.

Lawson, D., G. Tucker, C. A. Naesseth, C. Maddison, and Y. Whye Teh (2018). "Twisted Variational Sequential Monte Carlo". *Third workshop on Bayesian Deep Learning (NeurIPS)*.

Le, T. A., M. Igl, T. Rainforth, T. Jin, and F. Wood (2018). "Auto-Encoding Sequential Monte Carlo". In: *International Conference on Learning Representations*.

Lee, A. and N. Whiteley (2016). "Forest resampling for distributed sequential Monte Carlo". *Journal of statistical analysis and data mining*. 9(4): 230–248.

Lee, A., C. Yau, M. B. Giles, A. Doucet, and C. C. Holmes (2010). "On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods". *Journal of Computational and Graphical Statistics*. 10(4): 769–789.

Lin, M., R. Chen, and J. S. Liu (2013). "Lookahead strategies for sequential Monte Carlo". *Statistical Science*. 28(1): 69–94.

Lindsten, F. and A. Doucet (2016). "Pseudo-Marginal Hamiltonian Monte Carlo". arXiv.org, arXiv:1607.02516.

Lindsten, F., A. Johansen, C. A. Naesseth, B. Kirkpatrick, T. B. Schön, J. Aston, and A. Bouchard-Côté (2017). "Divide-and-Conquer with Sequential Monte Carlo". *Journal of Computational and Graphical Statistics*. 26(2): 445–458.

Lindsten, F., J. Helske, and M. Vihola (2018). "Graphical model inference: Sequential Monte Carlo meets deterministic approximations". In: *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc. 8201–8211.

Lindsten, F., M. I. Jordan, and T. B. Schön (2014). "Particle Gibbs with ancestor sampling". *The Journal of Machine Learning Research*. 15(1): 2145–2184.

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

Liu, J. S. (2004). *Monte Carlo strategies in scientific computing*. Springer Science+Business Media.

Louizos, C., U. Shalit, J. M. Mooij, D. Sontag, R. Zemel, and M. Welling (2017). “Causal Effect Inference with Deep Latent-Variable Models”. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc. 6446–6456.

MacEachern, S. N., M. Clyde, and J. S. Liu (1999). “Sequential importance sampling for nonparametric Bayes models: The next generation”. *Canadian Journal of Statistics*. 27(2): 251–267.

Maddison, C. J., D. Lawson, G. Tucker, N. Heess, M. Norouzi, A. Mnih, A. Doucet, and Y. Whye Teh (2017). “Filtering Variational Objectives”. In: *Advances in Neural Information Processing Systems*.

Meent, J.-W. van de, B. Paige, H. Yang, and F. Wood (2018). “An Introduction to Probabilistic Programming”. *arXiv preprint arXiv:1809.10756*.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). “Equation of state calculations by fast computing machines”. *The Journal of Chemical Physics*. 21(6): 1087–1092.

Metropolis, N. and S. Ulam (1949). “The Monte Carlo method”. *Journal of the American statistical association*. 44(247): 335–341.

Murray, I. and M. M. Graham (2016). “Pseudo-Marginal Slice Sampling”. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*. Cadiz, Spain.

Murray, L. M., A. Lee, and P. E. Jacob (2016). “Parallel resampling in the particle filter”. *Journal of Computational and Graphical Statistics*. 25(3): 789–805.

Naesseth, C. A., S. Linderman, R. Ranganath, and D. M. Blei (2018). “Variational Sequential Monte Carlo”. In: *International Conference on Artificial Intelligence and Statistics*. Vol. 84. PMLR. 968–977.

Naesseth, C. A., F. Lindsten, and T. B. Schön (2015). “Nested sequential Monte Carlo methods”. In: *International Conference on Machine Learning (ICML)*. 1292–1301.

Naesseth, C. A., F. Lindsten, and T. B. Schön (2019). “High-dimensional filtering using nested sequential Monte Carlo”. *IEEE Transactions on Signal Processing*. 67(16): 4177–4188.

Naesseth, C. A., F. Ruiz, S. W. Linderman, and D. M. Blei (2017). "Reparameterization gradients through acceptance-rejection sampling algorithms". In: *Artificial Intelligence and Statistics (AISTATS)*. 489–498.

Naesseth, C. A., F. Lindsten, and T. B. Schön (2014). "Sequential Monte Carlo for Graphical Models". In: *Advances in Neural Information Processing Systems 27*. Montreal, Canada: Curran Associates, Inc. 1862–1870.

Neal, R. M. (2001). "Annealed importance sampling". *Statistics and computing*. 11(2): 125–139.

Osendorfer, C. and J. Bayer (2014). "Learning stochastic recurrent networks". *Tech. rep.* arXiv:1411.7610.

Paige, B. and F. Wood (2016). "Inference networks for sequential Monte Carlo in graphical models". In: *International Conference on Machine Learning*. 3040–3049.

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

Rainforth, T., C. Naesseth, F. Lindsten, B. Paige, J.-W. Van de Meent, A. Doucet, and F. Wood (2016). "Interacting Particle Markov Chain Monte Carlo". In: *International Conference on Machine Learning*. 2616–2625.

Rebeschini, P. and R. Van Handel (2015). "Can local particle filters beat the curse of dimensionality?" *The Annals of Applied Probability*. 25(5): 2809–2866.

Rezende, D. J., S. Mohamed, and D. Wierstra (2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *International Conference on Machine Learning*. PMLR. 1278–1286.

Robbins, H. and S. Monro (Sept. 1951). "A Stochastic Approximation Method". *The Annals of Mathematical Statistics*. 22(3): 400–407.

Robert, C. and G. Casella (2004). *Monte Carlo statistical methods*. Springer Science & Business Media.

Rubin, D. B. (1987). "Comment: A Noniterative Sampling/Importance Resampling Alternative to the Data Augmentation Algorithm for Creating a Few Imputations When Fractions of Missing Information Are Modest: The SIR Algorithm". *Journal of the American Statistical Association*. 82(398): 543–546.

Ruiz, F., M. Titsias, and D. Blei (2016). "The generalized reparameterization gradient". In: *Advances in neural information processing systems*. 460–468.

Schäfer, C. and N. Chopin (2013). "Sequential Monte Carlo on large binary sampling spaces". *Statistics and Computing*. 23(2): 163–184.

Schön, T. B., A. Wills, and B. Ninness (Jan. 2011). "System Identification of Nonlinear State-Space Models". *Automatica*. 47(1): 39–49.

Snyder, C., T. Bengtsson, P. Bickel, and J. Anderson (2008). "Obstacles to high-dimensional particle filtering". *Monthly Weather Review*. 136(12): 4629–4640.

Snyder, C., T. Bengtsson, and M. Morzfeld (2015). "Performance Bounds for Particle Filters Using the Optimal Proposal". *Monthly Weather Review*. 143(11): 4750–4761.

Stewart, L. and P. McCarty Jr. (1992). "Use of Bayesian belief networks to fuse continuous and discrete information for target recognition, tracking, and situation assessment". In: *Proc. SPIE*. Vol. 1699. 177–185.

Van Der Merwe, R., A. Doucet, N. De Freitas, and E. A. Wan (2001). "The unscented particle filter". In: *Advances in neural information processing systems (NIPS)*. 584–590.

Vergé, C., C. Dubarry, P. Del Moral, and E. Moulines (2015). "On parallel implementation of sequential Monte Carlo methods: the island particle model". *Statistics and Computing*. 25(2): 243–260.

Wagner, W. (1987). "Unbiased Monte Carlo evaluation of certain functional integrals". *Journal of Computational Physics*. 71(1): 21–33.

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

Whiteley, N. (2013). "Stability properties of some particle filters". *The Annals of Applied Probability*. 23(6): 2500–2537.

Whiteley, N., A. Lee, and K. Heine (2016). "On the role of interaction in sequential Monte Carlo algorithms". *Bernoulli*. 22(1): 494–529.

Whitley, D. (1994). "A genetic algorithm tutorial". *Statistics and computing*. 4(2): 65–85.

Wood, F., J. W. Meent, and V. Mansinghka (2014). "A new approach to probabilistic programming inference". In: *Artificial Intelligence and Statistics*. 1024–1032.