Sequential Monte Carlo in the machine learning toolbox

Working with the trend of blending

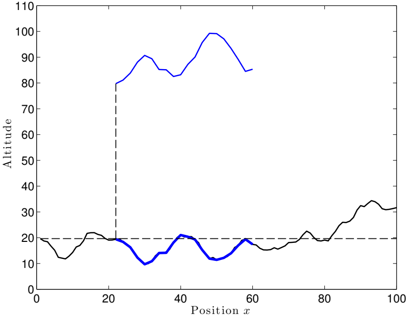
Thomas Schön
Uppsala University
Sweden.

Symposium on Advances in Approximate Bayesian Inference (AABI)
Montréal, Canada, December 2, 2018.

Sequential Monte Carlo – introductory toy example (I/II)

Consider a toy 1D localization problem.

Data



Model

Dynamic model:

$$x_{t+1} = x_t + u_t + v_t,$$

where x_t denotes position, u_t denotes velocity (known), $v_t \sim \mathcal{N}(0,5)$ denotes an unknown disturbance.

Measurements:

$$y_t = h(x_t) + e_t.$$

where $h(\cdot)$ denotes the world model (here the terrain height) and $e_t \sim \mathcal{N}(0,1)$ denotes an unknown disturbance.

Task: Find the state x_t (position) based on the measurements $y_{1:t} \triangleq \{y_1, \ldots, y_t\}$ by computing the filter density $p(x_t \mid y_{1:t})$.

Sequential Monte Carlo – introductory toy example (II/II)

Highlights two **key capabilities** of SMC:

1. Automatically handles an unknown and dynamically changing number of hypotheses.
2. Work with non-linear/non-Gaussian models.

Representation used: $p(x_t \mid y_{1:t}) = \sum_{i=1}^{N} \frac{w_t^i}{\sum_{l=1}^{N} w_t^l} \delta_{x_t^i}(x_t)$.

Aim and outline

Aim: To provide intuition for the **key mechanisms** underlying sequential Monte Carlo (SMC) and **hint at** a few ways in which SMC fits into the machine learning toolbox.

Outline:

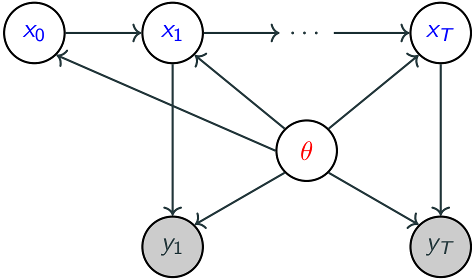
1. Introductory toy example
2. **SMC explained for dynamical systems**
3. SMC is a general method
4. "SMC should probably be blended"

Representing a nonlinear dynamical systems

The state space model is a **Markov** chain that makes use of a **latent** variable representation to describe dynamical phenomena.

Consists of the unobserved (state) process $\{x_t\}_{t\geq 0}$ modelling the dynamics and the observed process $\{y_t\}_{t\geq 1}$ modelling the relationship between the measurements and the unobserved state process:

$$x_t = f(x_{t-1}, \theta) + v_t,$$
$$y_t = g(x_t, \theta) + e_t.$$



Representations using distributions and programmatic models

Representation using probability distributions

$$x_t \mid (x_{t-1}, \theta) \sim p(x_t \mid x_{t-1}, \theta),$$
$$y_t \mid (x_t, \theta) \sim p(y_t \mid x_t, \theta),$$
$$x_0 \sim p(x_0 \mid \theta).$$

Representation using a programmatic model

x[1] ~ Gaussian(0.0, 1.0);	$p(x_1)$
y[1] ~ Gaussian(x[1], 1.0);	$p(y_1 \mid x_1)$
for (t in 2..T) {	
x[t] ~ Gaussian(a*x[t - 1], 1.0);	$p(x_t \mid x_{t-1})$
y[t] ~ Gaussian(x[t], 1.0);	$p(y_t \mid x_t)$
}	

A **probabilistic program** encodes a **probabilistic model** using a particular probabilistic programming language (here Birch).

Lawrence Murray and Thomas B. Schön. **Automated learning with a probabilistic programming language: Birch**. *Annual Reviews in Control*, 2018. (To appear)

State space model – full probabilistic model

The **full probabilistic model** is given by

$$p(x_{0:T}, \theta, y_{1:T}) = \underbrace{\prod_{t=1}^{T} \underbrace{p(y_t \mid x_t, \theta)}_{\text{observation}}}_{\text{likelihood } p(y_{1:T} \mid x_{0:T}, \theta)} \underbrace{\prod_{t=1}^{T} \underbrace{p(x_t \mid x_{t-1}, \theta)}_{\text{dynamics}} \underbrace{p(x_0 \mid \theta)}_{\text{state}} \underbrace{p(\theta)}_{\text{param.}}}_{\text{prior } p(x_{0:T}, \theta)}$$

The **nonlinear filtering problem** involves the measurement update

$$p(x_t \mid y_{1:t}) = \frac{\overbrace{p(y_t \mid x_t)}^{\text{measurement}} \overbrace{p(x_t \mid y_{1:t-1})}^{\text{prediction pdf}}}{p(y_t \mid y_{1:t-1})},$$

and the time update

$$p(x_t \mid y_{1:t-1}) = \int \underbrace{p(x_t \mid x_{t-1})}_{\text{dynamics}} \underbrace{p(x_{t-1} \mid y_{1:t-1})}_{\text{filtering pdf}} \mathrm{d}x_{t-1}.$$

Sequential Monte Carlo (SMC)

The need for approximate methods (such as SMC) is tightly coupled to the intractability of the integrals above.

SMC provide approximate solutions to **integration** problems where there is a **sequential structure** present.

The **particle filter** approximates $p(x_t \mid y_{1:t})$ for

$$x_t = f(x_{t-1}) + v_t,$$
$$y_t = g(x_t) + e_t,$$

by maintaining an **empirical distribution** made up of N samples (particles) $\{x_t^i\}_{i=1}^N$ and the corresponding weights $\{w_t^i\}_{i=1}^N$

$$\underbrace{\widehat{p}(x_t \mid y_{1:t})}_{\widehat{\pi}(x_t)} = \sum_{i=1}^{N} \frac{w_t^i}{\sum_{l=1}^{N} w_t^l} \delta_{x_t^i}(x_t).$$

SMC – the particle filter in its simplest form

SMC = sequential importance sampling + resampling

1. **Propagation:** $x_t^i \sim p(x_t \mid x_{1:t-1}^{a_t^i})$ and $x_{1:t}^i = \{x_{1:t-1}^{a_t^i}, x_t^i\}$.
2. **Weighting:** $w_t^i = W_t(x_t^i) = p(y_t \mid x_t^i)$.
3. **Resampling:** $\mathbb{P}\big(a_t^i = j\big) = w_{t-1}^j / \sum_l w_{t-1}^l$.

The **ancestor indices** $\{a_t^i\}_{i=1}^N$ are very **useful** auxiliary variables!
They make the stochasticity of the resampling step explicit.

SMC – in words

1. **Propagation:** Sample a new successor state and append it to the earlier.
2. **Weighting:** The weights corrects for the discrepancy between the proposal distribution and the target distribution.
3. **Resampling:** Focus the computation on the promising parts of the state space by randomly pruning particles, while still preserving the asymptotic guarantees of importance sampling.

Application – indoor localization using the magnetic field (I/II)

Aim: Compute the **position** using variations in the ambient magnetic field and the motion of the person (acceleration and angular velocities). All of this observed using sensors in a standard smartphone.

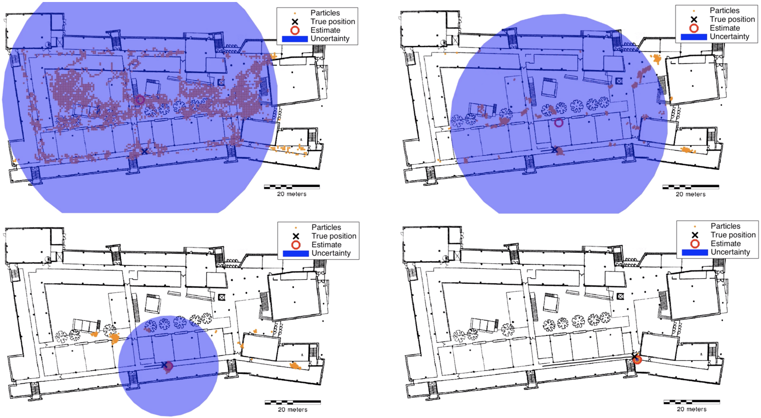
First we need a map, which we build using a tailored Gaussian process.

www.youtube.com/watch?v=enlMiUqPVJo

Arno Solin, Manon Kok, Niklas Wahlström, TS and Simo Särkkä. **Modeling and interpolation of the ambient magnetic field by Gaussian processes**. *IEEE Transactions on Robotics*, 34(4):1112–1127, 2018.

Carl Jidling, Niklas Wahlström, Adrian Wills and TS. **Linearly constrained Gaussian processes**. *Advances in Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, December, 2017.

Application – indoor localization using the magnetic field (II/II)



Show movie!

Arno Solin, Simo Särkkä, Juho Kannala and Esa Rahtu. **Terrain navigation in the magnetic landscape: Particle filtering for indoor positioning.** In *Proceedings of the European Navigation Conference*, Helsinki, Finland, June, 2016.

Sequential Monte Carlo (SMC) – abstract

The distribution of interest $\pi(x)$ is called the **target distribution**.

(Abstract) problem formulation: **Sample from a sequence** of probability distributions $\{\pi_t(x_{0:t})\}_{t\geq 1}$ defined on a sequence of spaces of increasing dimension, where

$$\pi_t(x_{0:t}) = \frac{\widetilde{\pi}_t(x_{0:t})}{Z_t},$$

such that $\widetilde{\pi}_t(x_t) : \mathcal{X}^t \to \mathbb{R}^+$ is known point-wise and $Z_t = \int \pi(x_{0:t})\mathrm{d}x_{0:t}$ is often computationally challenging.

SMC methods are a class of sampling-based algorithms capable of:

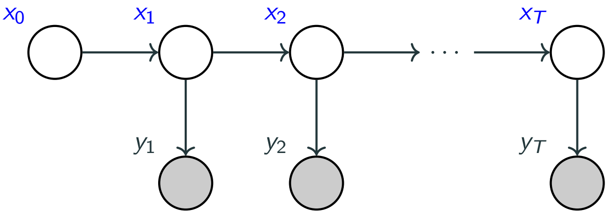
1. Approximating $\pi(x)$ and compute integrals $\int \varphi(x)\pi(x)\mathrm{d}x$.
2. Approximating the normalizing constant Z (unbiased).

Important question: How general is this formulation?

SMC is actually more general than we first thought

The sequence of target distributions $\{\pi_t(x_{1:t})\}_{t=1}^n$ can be constructed in **many** different ways.

The most basic construction arises from **chain-structured graphs**, such as the state space model.



$$\overbrace{p(x_{1:t} \mid y_{1:t})}^{\pi_t(x_{1:t})} = \frac{\overbrace{p(x_{1:t}, y_{1:t})}^{\widetilde{\pi}_t(x_{1:t})}}{\underbrace{p(y_{1:t})}_{Z_t}}$$

SMC can be used for general graphical models

SMC methods are used to approximate a **sequence of probability distributions** on a sequence of spaces of increasing dimension.

Key idea:

1. Introduce a **sequential decomposition** of any probabilistic graphical model.
2. Each **subgraph** induces an intermediate target dist.
3. Apply SMC to the sequence of intermediate target dist.

SMC also provides an unbiased estimate of the **partition function**!

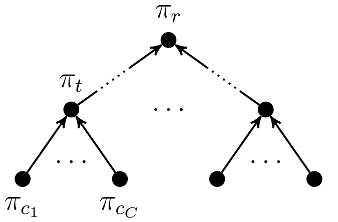
Christian A. Naesseth, Fredrik Lindsten and TS. **Sequential Monte Carlo methods for graphical models**. In *Advances in Neural Information Processing Systems (NIPS)* 27, Montreal, Canada, December, 2014.

Going from classical SMC fo D&C-SMC

The **computational graph** of classic SMC is a sequence (chain)

Iteration:

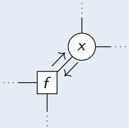
D&C-SMC generalize the classical SMC framework **from sequences to trees**.



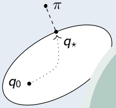
Fredrik Lindsten, Adam M. Johansen, Christian A. Naesseth, Bonnie Kirkpatrick, TS, John Aston and Alexandre Bouchard-Côté. **Divide-and-Conquer with Sequential Monte Carlo**. *Journal of Computational and Graphical Statistics (JCGS)*, 26(2):445-458, 2017.

Deterministic methods

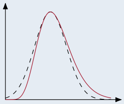
Message passing

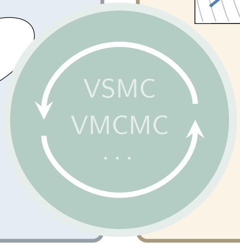


Variational inf.



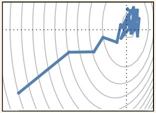
Laplace's method



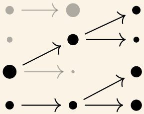


Monte Carlo methods

Markov chain Monte Carlo



Sequential Monte Carlo



Blending deterministic and Monte Carlo methods

Deterministic methods:

Good: Accurate and rapid inference

Bad: Results in biases that are hard to quantify

Monte Carlo methods:

Good: Asymptotic consistency, lots of theory available

Bad: Can suffer from a high computational cost

Examples of freedom in the SMC algorithm that opens up for **blending**:

The **proposal** distributions can be defined in many ways.

The **intermediate target** distributions can be defined in many ways.

Leads to very interesting and useful algorithms.

Recent developments working with the trend of blending

Develop new approximating families of distributions.

Naesseth, C. A., Linderman, S. W., Ranganath, R. and Blei, D. M. **Variational Sequential Monte Carlo**. *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

Maddison, C. J., Lawson, D., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A. and Teh, Y. W. **Filtering variational objectives**. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Le, T. A., Igl, M., Rainforth, T., Jin, T. and Wood, F. **Auto-encoding sequential Monte Carlo**. In *International Conference on Learning Representations (ICLR)*, 2018.

Alter the intermediate targets to take "future variables" into account.

Results in **"additional intractability"** – use deterministic methods.

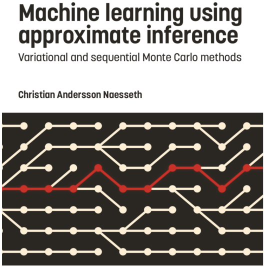
Alternative interpretation: Use SMC as a post-correction for the bias introduced by the deterministic methods.

Lindsten, F., Helske, J. and Vihola, M. **Graphical model inference: Sequential Monte Carlo meets deterministic approximations**. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

"The combination of the two ideas mentioned above".

Lawson, D., Tucker, G., Naesseth, C. A., Maddison, C. J., Adams, R. P., and Teh, Y. W. **Twisted Variational Sequential Monte Carlo**. *Bayesian Deep Learning (NeurIPS Workshop)*, 2018.

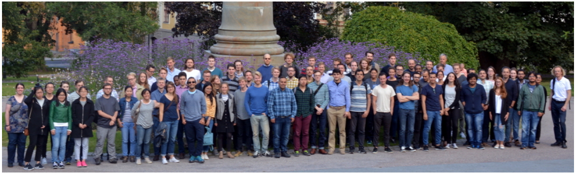
New PhD thesis and a PhD level course on the topic



Christian A. Naesseth will defend his PhD thesis on this topic next week:

users.isy.liu.se/en/rt/chran60/index.html

Intensive PhD course on **SMC methods** in August 2019. **Welcome!**



www.it.uu.se/research/systems_and_control/education/2019/smc

Conclusion

SMC provide approximate solutions to **integration** problems where there is a **sequential structure** present.

- SMC is **more general** than we first though.
- SMC can indeed be **computationally challenging**, but it comes with rather well-developed analysis and guarantees.
- There is still a lot of freedom **waiting to be exploited**.

Forthcoming SMC introduction written with an ML audience in mind

Christian A. Naesseth, Fredrik Lindsten, and TS. **Elements of sequential Monte Carlo**. *Foundations and Trends in Machine Learning*, 2019 (initial draft available).