Ancestor sampling in state space models and beyond

“Ancestor sampling is a way of exploiting backward simulation ideas without needing an explicit backward pass.”

Thomas Schön
Division of Systems and Control
Department of Information Technology
Uppsala University, Sweden

Joint work with (alphabetical order): Michael I. Jordan (UC Berkeley), Fredrik Lindsten (University of Cambridge) and Christian A. Naesseth (Linköping University).
Dynamical systems are everywhere!

Some of the dynamical systems we have been working with,

We first have to learn the models. Then we can use them.
A state space model (SSM) consists of a Markov process \( \{x_t\}_{t \geq 1} \)
and a measurement process \( \{y_t\}_{t \geq 1} \), related according to

\[
\begin{align*}
    x_{t+1} \mid x_t & \sim f_t(x_{t+1} \mid x_t), \\
    y_t \mid x_t & \sim g_t(y_t \mid x_t), \\
    x_1 & \sim \mu(x_1).
\end{align*}
\]

We observe

\[
y_{1:T} \triangleq \{y_1, \ldots, y_T\},
\]

(leading the latent variables \( x_{1:T} \) unobserved).

**Learning problem**: Find \( f, g, \mu \) (or \( \theta \)) based on \( y_{1:T} \).
Alternate between updating $\theta$ and updating $x_{1:T}$.

Frequentists:
- Find $\hat{\theta}_{\text{ML}} = \arg \max_{\theta} p_{\theta}(y_{1:T})$.
- Use e.g. the expectation maximization (EM) algorithm.

Bayesians:
- Find $p(\theta \mid y_{1:T})$.
- Use e.g. Gibbs sampling.

Frequentists + Bayesians:
- Use e.g. particle Gibbs together with stochastic approx. EM.
1. Maximum Likelihood (ML) learning
   a) Problem formulation
   b) EM and a particle smoother
2. Bayesian learning
   a) Problem formulation
   b) Gibbs sampling
3. Sequential Monte Carlo (SMC)
4. Particle Gibbs with ancestor sampling (PGAS)
5. SMC and PGAS for graphical models

The sequential Monte Carlo samplers are fundamental to both the ML and the Bayesian approaches.

Thomas Schönen, Ancestor sampling in state space models and beyond
A state space model (SSM) consists of a Markov process \( \{x_t\}_{t \geq 1} \) and a measurement process \( \{y_t\}_{t \geq 1} \), related according to

\[
\begin{align*}
    x_{t+1} | x_t & \sim f_{\theta,t}(x_{t+1} | x_t), \\
    y_t | x_t & \sim g_{\theta,t}(y_t | x_t), \\
    x_1 & \sim \mu_{\theta}(x_1).
\end{align*}
\]

**Learning problem:** Find \( \theta \) based on \( y_{1:T} \).

ML amounts to solving,

\[
\hat{\theta}^{\text{ML}} = \arg \max_{\theta} \log p_{\theta}(y_{1:T})
\]

where the log-likelihood function is given by

\[
\log p_{\theta}(y_{1:T}) = \sum_{t=1}^{T} \log p_{\theta}(y_t | y_{1:t-1})
\]
Algorithm 1 EM – learning nonlinear dynamical systems

1. **Initialise**: Set $i = 1$ and choose an initial $\theta^1$.
2. **While** not converged **do**:

   (a) **Expectation (E) step**: Compute

   \[
   Q(\theta, \theta^i) = E_{\theta^i} \left[ \log p_\theta(x_{1:T}, y_{1:T}) \mid y_{1:T} \right]
   \]

   \[
   = \int \log p_\theta(x_{1:T}, y_{1:T}) p_{\theta^i}(x_{1:T} \mid y_{1:T}) \, dx_{1:T}
   \]

   using **PS** (backward simulation or ancestor sampling).

   (b) **Maximization (M) step**: Compute $\theta^{i+1} = \arg \max_{\theta \in \Theta} Q(\theta, \theta^i)$

   (c) $i \leftarrow i + 1$

---


Thomas Schön (user.it.uu.se/thosc112), *Ancestor sampling in state space models and beyond*

Consider a Bayesian SSM ($\theta$ is now a r.v. with a prior density $p(\theta)$)

$$x_{t+1} \mid x_t \sim f_{\theta,t}(x_{t+1} \mid x_t),$$
$$y_t \mid x_t \sim g_{\theta,t}(y_t \mid x_t),$$
$$x_1 \sim \mu_\theta(x_1),$$
$$\theta \sim p(\theta).$$

**Learning problem:** Compute the posterior $p(\theta, x_{1:T} \mid y_{1:T}),$ or one of its marginals.

**Key challenge:** There is no closed form expression available.
**Gibbs sampler for SSMs (I/II)**

**Aim:** Compute $p(\theta, x_{1:T} \mid y_{1:T})$.

**MCMC:** Gibbs sampling (blocked) for SSMs amounts to iterating

- Draw $\theta[m] \sim p(\theta \mid x_{1:T}[m-1], y_{1:T})$,
- Draw $x_{1:T}[m] \sim p(x_{1:T} \mid \theta[m], y_{1:T})$.

The above procedure results in a Markov chain,

$$\{\theta[m], x_{1:T}[m]\}_{m \geq 1}$$

with $p(\theta, x_{1:T} \mid y_T)$ as its stationary distribution!
Gibbs sampler for SSMs (II/II)

**Aim:** Compute $p(\theta, x_{1:T} \mid y_{1:T})$.

**MCMC:** Gibbs sampling (blocked) for SSMs amounts to iterating

- Draw $\theta[m] \sim p(\theta \mid x_{1:T}[m-1], y_{1:T})$; **OK!**
- Draw $x_{1:T}[m] \sim p(x_{1:T} \mid \theta[m], y_{1:T})$. **Hard!**

**Problem:** $p(x_{1:T} \mid \theta[m], y_{1:T})$ not available!

**Idea:** Approximate $p(x_{1:T} \mid \theta[m], y_{1:T})$ using a sequential Monte Carlo method!
1. Maximum Likelihood (ML) learning
   a) Problem formulation
   b) Solution using EM and a particle smoother
2. Bayesian learning
   a) Problem formulation
   b) Gibbs sampling
3. **Sequential Monte Carlo (SMC)**
4. Particle Gibbs with ancestor sampling (PGAS)
5. SMC and PGAS for graphical models

The sequential Monte Carlo samplers are fundamental to both the ML and the Bayesian approaches.
Sequential Monte Carlo (SMC)

Approximate a **sequence** of probability distributions on a sequence of probability spaces of **increasing dimension**.

Let \( \{ \gamma_{\theta,t}(x_{1:t}) \}_{t \geq 1} \) be a sequence of unnormalized densities and

\[
\tilde{\gamma}_{\theta,t}(x_{1:t}) = \frac{\gamma_{\theta,t}(x_{1:t})}{Z_{\theta,t}}
\]

**Ex. (SSM)**

\[
\tilde{\gamma}_{\theta,t}(x_{1:t}) = p_{\theta}(x_{1:t} \mid y_{1:t}), \quad \gamma_{\theta,t}(x_{1:t}) = p_{\theta}(x_{1:t}, y_{1:t}),
\]

\[
Z_{\theta,t} = p_{\theta}(y_{1:t}).
\]
Consider a toy 1D localization problem.

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.

The same idea has been used for the Swedish fighter JAS 39 Gripen. Details are available in,

Highlights two key capabilities of SMC:

1. Automatically handles an unknown and dynamically changing number of hypotheses.

2. Work with nonlinear/non-Gaussian models.

\[ p(x_t \mid y_{1:t}) \approx \sum_{i=1}^{N} w_i \delta_{x_i^t}(x_t) \]
Sequential Monte Carlo (SMC)

SMC = resampling + sequential importance sampling

1. **Resampling:** \( \mathbb{P}(a_t^i = j) = \frac{w_{t-1}^j}{\sum_l w_{t-1}^l} \).

2. **Propagation:** \( x_t^i \sim r_{\theta,t}(x_t \mid x_{1:t-1}^{a_t^i}) \) and \( x_{1:t}^i = \{ x_{1:t-1}^{a_t^i}, x_t^i \} \).

3. **Weighting:** \( w_t^i = W_{\theta,t}(x_{1:t}^i) = \frac{\gamma_{\theta,t}(x_{1:t}^i)}{\gamma_{\theta,t-1}(x_{1:t-1}^{a_t^i}) r_{\theta,t}(x_t^i \mid x_{1:t-1}^{a_t^i})} \).

The result is a new weighted set of particles \( \{ x_{1:t}^i, w_t^i \}_{i=1}^N \).
Algorithm SMC (each step is for \( i = 1, \ldots N \))

1. **Initialize** \( (t = 1) \):
   
   (a) Draw \( x_1^i \sim r_{\theta,1}(x_1^i) \).
   
   (b) Set \( w_1^i = W_{\theta,1}(x_1^i) \).

2. **For** \( t = 2 \) to \( T \) **do**:
   
   (a) Draw \( a_t^i \sim \text{Cat}(\{w_t^j\}_{j=1}^N) \).
   
   (b) Draw \( x_t^i \sim r_{\theta,t}(x_t | x_{1:t-1}^{a_t^i}) \) and set \( x_{1:t}^i = \{x_{1:t-1}^{a_t^i}, x_t^i\} \).
   
   (c) Set \( w_t^i = W_{\theta,t}(x_{1:t}^i) \).
SMC suffers from path degeneracy (illustration)
**Aim:** Compute the position of a person moving around indoors using sensors (inertial, magnetometer and radio) located in an ID badge and a map.
1. Maximum Likelihood (ML) learning
   a) Problem formulation
   b) Solution using EM and a particle smoother
2. Bayesian learning
   a) Problem formulation
   b) Gibbs sampling
3. Sequential Monte Carlo (SMC)
4. Particle Gibbs with ancestor sampling (PGAS)
5. SMC and PGAS for graphical models

The sequential Monte Carlo samplers are fundamental to both the ML and the Bayesian approaches.
Recall – Gibbs sampler for SSMs

**Aim:** Compute \( p(\theta, x_{1:T} | y_{1:T}) \).

**MCMC:** Gibbs sampling (blocked) for SSMs amounts to iterating

- Draw \( \theta[m] \sim p(\theta | x_{1:T}[m-1], y_{1:T}) \); \( \text{OK!} \)
- Draw \( x_{1:T}[m] \sim p(x_{1:T} | \theta[m], y_{1:T}) \). \( \text{Hard!} \)

**Problem:** \( p(x_{1:T} | \theta[m], y_{1:T}) \) not available!

**Idea:** Approximate \( p(x_{1:T} | \theta[m], y_{1:T}) \) using a sequential Monte Carlo method!
With \( \mathbb{P}(x_{1:T}^* = x_{1:T}^i) \propto w_T^i \) we get, \( x_{1:T}^* \approx p(x_{1:T} | \theta, y_{1:T}) \).
Problems with this approach,

- Based on a PF ⇒ approximate sample.
- Does not leave $p(x_{1:T} \mid \theta, y_{1:T})$ invariant!
- Relies on large $N$ to be successful.
- A lot of wasted computations.

To get around these problems,

Use a conditional particle filter (CPF). One pre-specified reference trajectory is retained throughout the sampler.


Thomas Schön (user.it.uu.se/thosc112), *Ancestor sampling in state space models and beyond*
The idea underlying PMCMC is to make use of a certain SMC sampler to construct a Markov kernel leaving the joint smoothing distribution $p(x_{1:T} \mid \theta, y_{1:T})$ invariant.

This Markov kernel is then used in a standard MCMC algorithm (e.g. Gibbs, results in the Particle Gibbs (PG)).

For a self-contained introduction (focused on BS and AS),

Three SMC samplers leaving $p(x_{1:T} \mid \theta, y_{1:T})$ invariant:

1. **Conditional particle filter (CPF)**


2. **CPF with backward simulation (CPF-BS)**


3. **CPF with ancestor sampling (CPF-AS)**


Let \( x'_{1:T} = (x'_1, \ldots, x'_T) \) be a fixed reference trajectory.

- At each time \( t \), sample only \( N - 1 \) particles in the standard way.
- Set the \( N^{\text{th}} \) particle deterministically: \( x_t^N = x'_t \).

CPF causes us to degenerate to the something that is very similar to the reference trajectory, resulting in slow mixing.
BS is problematic for models with more intricate dependencies.

**Reason:** Requires complete trajectories of the latent variable in the backward sweep.

**Solution:** Modify the computation to achieve the same effect as BS, but without an explicit backwards sweep.

**Implication:** Ancestor sampling opens up for inference in a wider class of models, e.g. non-Markovian SSMs, PGMs and BNP models.

Ancestor sampling is conceptually similar to backward simulation, but instead of separate forward and backward sweeps, we achieve the same effect in a **single forward sweep**.
Let $x'_{1:T} = (x'_1, \ldots, x'_T)$ be a fixed reference trajectory.

- At each time $t$, sample only $N - 1$ particles in the standard way.
- Set the $N^{th}$ particle deterministically: $x^N_t = x'_t$.
- Generate an artificial history for $x^N_t$ by ancestor sampling.

CPF-AS causes us to degenerate to something that is very different from the reference trajectory, resulting in better mixing.
Algorithm CPF-AS, conditioned on $x'_{1:T}$

1. **Initialize** ($t = 1$):
   (a) Draw $x^i_1 \sim r_{\theta,1}(x^i_1)$, for $i = 1, \ldots, N - 1$.
   (b) Set $x^N_1 = x'_1$.
   (c) Set $w^i_1 = W_{\theta,1}(x^i_1)$.

2. **For** $t = 2$ **to** $T$ **do**:
   (a) Draw $a^i_t \sim \text{Cat}(\{w^j_t\}_{j=1}^N)$, for $i = 1, \ldots, N - 1$.
   (b) Draw $x^i_t \sim r_{\theta,t}(x_t | x^a^i_{1:t-1})$, for $i = 1, \ldots, N - 1$.
   (c) Set $x^N_t = x'_t$.
   (d) Draw $a^N_t$ with $\mathbb{P}(a^N_t = i) \propto w^i_{t-1} \frac{\gamma_{\theta,T}(x^i_{1:t-1}, x'_t)}{\gamma_{\theta,t-1}(x^i_{1:t-1})}$.
   (e) Set $x^i_{1:t} = \{x^a^i_{1:t-1}, x^i_t\}$ and $w^i_t = W_{\theta,t}(x^i_{1:t})$. 
The PGAS Markov kernel

1. Run CPF-AS($x'_{1:T}$) targeting $p(x_{1:T} \mid \theta, y_{1:T})$.
2. Sample $x^*_{1:T}$ with $\mathbb{P}(x^*_{1:T} = x^i_{1:T}) \propto w^i_T$.

- Maps $x'_{1:T}$ stochastically into $x^*_{1:T}$
- Implicitly defines an ergodic Markov kernel ($P_N^\theta$) referred to as the PGAS (particle Gibbs with ancestor sampling) kernel.

**Theorem**

For any number of particles $N \geq 1$ and $\theta \in \Theta$, the PGAS kernel $P_N^\theta$ leaves $p(x_{1:T} \mid \theta, y_{1:T})$ invariant,

$$p(dx^*_{1:T} \mid \theta, y_{1:T}) = \int P_N^\theta(x'_{1:T}, dx^*_{1:T})p(dx'_{1:T} \mid \theta, y_{1:T})$$
Bayesian learning: Gibbs + CPF-AS = PGAS

Algorithm Particle Gibbs with ancestor sampling (PGAS)

1. **Initialize:** Set $\{\theta[0], x_{1:T}[0]\}$ arbitrarily.

2. **For $m \geq 1$, iterate:**
   
   (a) Draw $\theta[m] \sim p(\theta \mid x_{1:T}[m-1], y_{1:T})$.
   
   (b) Run CPF-AS($x_{1:T}[m-1]$), targeting $p(x_{1:T} \mid \theta[m], y_{1:T})$.
   
   (c) Sample with $P(x_{1:T}[m] = x_{1:T}^i) \propto w_T^i$. 
Consider the stochastic volatility model,

\[ x_{t+1} = 0.9 x_t + w_t, \quad w_t \sim \mathcal{N}(0, \theta), \]

\[ y_t = e_t \exp \left( \frac{1}{2} x_t \right), \quad e_t \sim \mathcal{N}(0, 1). \]

Let us study the ACF for the estimation error, \( \hat{\theta} - \mathbb{E} [\theta | y_{1:T}] \).
Plots of the update rate of $x_t$ versus $t$, i.e. the proportion of iterations where $x_t$ changes value. This provides another comparison of the mixing.
Two more PGAS application examples

1. A Gaussian process state space model (GP-SSM)

\[ f(x_t) \sim \mathcal{GP}(m_{\theta_x}(x_t), k_{\theta_x}(x_t, x'_t)), \]
\[ x_{t+1} | f_t \sim \mathcal{N}(x_{t+1} | f_t, Q), \]
\[ y_t | x_t \sim p(y_t | x_t, \theta_y). \]

**Key idea:** Marginalize out the function \( f \) using PGAS.


2. ML learning using PGAS + stochastic approximation EM (PSAEM)

\[
\hat{Q}_m(\theta) = \hat{Q}_{m-1}(\theta) + \gamma_m \left( \log p_\theta(x_{1:T}[m], y_{1:T}) - \hat{Q}_{m-1}(\theta) \right)
\]


Thomas Schön (user.it.uu.se/thosc112), *Ancestor sampling in state space models and beyond*
Outline

1. Maximum Likelihood (ML) learning
   a) Problem formulation
   b) Solution using EM and a particle smoother
2. Bayesian learning
   a) Problem formulation
   b) Gibbs sampling
3. Sequential Monte Carlo (SMC)
4. Particle Gibbs with ancestor sampling (PGAS)
5. SMC and PGAS for graphical models

The sequential Monte Carlo samplers are fundamental to both the ML and the Bayesian approaches.
Constructing an artificial sequence of intermediate (auxiliary) target distributions for an SMC sampler is a powerful (and quite possibly underutilized) idea. For some applications, see


---

**Key idea:** Perform and make use of a sequential decomposition of the probabilistic graphical model (PGM).

Defines a sequence of intermediate (auxiliary) target distributions defined on an increasing sequence of probability spaces.

Target this sequence using SMC.
The joint PDF of the set of random variables indexed by $\mathcal{V}$, $X_\mathcal{V} \triangleq \{x_1, \ldots, x_{|\mathcal{V}|}\}$

$$p(X_\mathcal{V}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(X_C).$$

Sequential decomposition of the above factor graph (the target distributions are built up by adding factors at each iteration),

$$\gamma_1(X_{\mathcal{L}_1})$$

$$\gamma_2(X_{\mathcal{L}_2})$$
Let \( \{\psi_k\}_{k=1}^K \) be a sequence of factors,

\[
\psi_k(X_{I_k}) = \prod_{C \in \mathcal{C}_k} \psi_C(X_C),
\]

where \( I_k \subseteq \{1, \ldots, |\mathcal{V}|\} \).

The **sequential decomposition** is based on these factors,

\[
\gamma_k(X_{L_k}) \triangleq \prod_{\ell=1}^k \psi_\ell(X_{I_\ell}),
\]

where \( L_k \triangleq \bigcup_{\ell=1}^k I_\ell \).

By construction, \( L_K = \mathcal{V} \) and the joint PDF \( p(X_{L_K}) \propto \gamma_K(X_{L_K}) \).
Algorithm SMC sampler for PGMs

1. **Initialize** \((k = 1)\): Draw \(X^i_{L_1} \sim r_1(\cdot)\) and set \(w^i_1 = W_1(X^i_{L_1})\).

2. **For** \(k = 2\) **to** \(K\) **do:**
   
   (a) Draw \(a^i_k \sim \text{Cat}(\{w^j_{k-1}\}_{j=1}^N)\).
   
   (b) Draw \(\xi^i_k \sim r_k(\cdot|X_{L_{k-1}}^{a^i_k})\) and set \(X^i_{L_k} = X_{L_{k-1}}^{a^i_k} \cup \xi^i_k\).
   
   (c) Set \(w^i_k = W_k(X^i_{L_k})\).

Also provides an estimate of the **partition function**!

**Problem:** SMC is not enough since:

1. It does not solve the parameter learning problem.
2. The quality of the marginals \(p(X_{L_k})\) deteriorates for \(k \ll K\).

**Solution:** Use PGAS.
Consider a standard squared lattice Gaussian MRF of size $10 \times 10$,

$$p(X_{\mathcal{V}}, Y_{\mathcal{V}}) \propto \prod_{i \in \mathcal{V}} e^{\frac{1}{2\sigma_i^2}(x_i - y_i)^2} \prod_{(i,j) \in \mathcal{E}} e^{\frac{1}{2\sigma_{ij}^2}(x_i - x_j)^2}$$

Four MCMC samplers:
1. PGAS – fully blocked
2. PGAS – partially blocked
3. Standard one-at-a-time Gibbs
4. Tree sampler (Hamze & de Freitas, 2004)

The arrows show the order in which the factors are added.
Example – Gaussian MRF (II/II)

Thomas Schön (user.it.uu.se/thosc112), *Ancestor sampling in state space models and beyond*
We have introduced several SMC-based inference methods for PGMs of arbitrary topologies with discrete or continuous variables. The sequential decomposition is **not unique** and its form will affect

- accuracy
- computational efficiency
- simplicity of implementation

Details and a loopy, non-Gaussian and non-discrete PGM example,

Conclusions

- Think of the PGAS kernel as a component that can be used in different inference algorithms.
- Not at all limited to SSMs. Particularly useful for models with more complex dependencies, such as
  - Non-Markovian models
  - Bayesian nonparametric models
  - Probabilistic graphical models
- PGAS is built upon two main ideas
  1. Conditioning the underlying SMC sampler on a reference trajectory ensures the correct stationary distribution for any $N$.
  2. Ancestor sampling causes degeneration to different trajectories, drastically improving the mixing of the sampler.

There is a lot of interesting research that remains to be done!!

Thomas Schön (user.it.uu.se/ thosc112), Ancestor sampling in state space models and beyond
Novel introduction of PMCMC (given us lots of inspiration)


Forthcoming book

Thomas B. Schön and Fredrik Lindsten, *Learning of dynamical systems – Particle filters and Markov chain methods*, 2014 (or 2015...).

Self-contained introduction to BS and AS (not limited to SSMs)


PGAS


Thomas Schön (user.it.uu.se/ thosc112), *Ancestor sampling in state space models and beyond*
ML identification of nonlinear SSMs


Bayesian inference in SSMs using Gaussian processes


SMC methods for graphical models


Some MATLAB code is available from the web-site.