Learning nonlinear dynamical models using particle filters

Part 4 – Identification strategy 2 (Data augmentation)

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

Email: thomas.schon@it.uu.se,
www: user.it.uu.se/~thosc112


Outline – Part 4

Aim: Show how SMC can be used to implement identification strategy 2 – Data augmentation.

1. Summary of Part 3
2. Identification strategy 2 – Data augmentation
3. Maximum likelihood via Expectation Maximization (EM)
4. Sampling state trajectories using Markov kernels
5. Bayesian identification using Gibbs sampling
6. Some of our current research activities (if there is time)

Summary of Part 3 (I/II)

Exact approximation: The stationary distribution of the Markov chain is the exact target distribution \( p(\theta | y_{1:T}) \) despite the use of an approximate computation of the acceptance probability in MH.

The approximate computation amounts to the use of SMC (with a finite number of particles) in estimating the likelihood (unbiasedly).

The idea underlying particle MCMC (PMCMC) is to make use of SMC algorithms to propose (simulate) state trajectories \( x_{1:T} \).

These state trajectories are then used within standard MCMC algorithms operating on non-standard spaces.

1. Particle Metropolis Hastings
2. Particle Gibbs (this lecture)

Summary of Part 3 (II/II)

We introduced an auxiliary variable

\[ u = (x_{1:T}, a_{2:T}), \quad u \sim \psi(\theta | y_{1:T}). \]

and considered an extended target distribution

\[ \phi(\theta, u | y_{1:T}) = \frac{\hat{p}_{\theta,u}(y_{1:T})\psi(u | y_{1:T})p(\theta)}{p(y_{1:T})}. \]

Could construct a standard MH algorithm that operates in the non-standard extended space \( \Theta \times X^{NT} \times \{1, \ldots, N\}^{N(T-1)} \).

The posterior \( p(\theta | y_{1:T}) \) is recovered exactly as the marginal of the extended target distribution \( \phi(\theta, u | y_{1:T}) \), despite the fact that we employ an SMC approximation of the likelihood using a finite number of particles \( N \).
Identification strategy 2 – data augmentation

Motivation: If we had access to the complete likelihood
\[ p_\theta(x_1:T, y_1:T) = \mu_\theta(x_1) \prod_{t=1}^{T} g_\theta(y_t | x_t) \prod_{t=1}^{T-1} f_\theta(x_{t+1} | x_t) \]
the problem would be much easier.

Key idea: Treat the state sequence \( x_1:T \) as an auxiliary variable that is estimated together with the parameters \( \theta \).

The data augmentation strategy breaks the original problem into two new and closely linked problems.

Intuitively: Alternate between updating \( \theta \) and \( x_1:T \).

Data augmentation – EM

Maximum likelihood formulation – model the unknown parameters as a deterministic variable and solve
\[ \hat{\theta}_{\text{ML}} = \arg \max_{\theta \in \Theta} p_\theta(y_1:T). \]

The expectation maximization algorithm is an iterative approach to compute ML estimates of unknown parameters (\( \theta \)) in probabilistic models involving latent variables (the state trajectory \( x_1:T \)).

Expectation maximization (EM) employs the complete likelihood \( p_\theta(x_1:T, y_1:T) \) as a substitute for the observed likelihood \( p_\theta(y_1:T) \),
\[ p_\theta(x_1:T, y_1:T) = p_\theta(x_1:T | y_1:T) p_\theta(y_1:T). \]

Data augmentation – EM

EM works by iteratively computing
\[ Q(\theta, \theta_k) = \int \log p_\theta(x_1:T, y_1:T)p_{\theta_k}(x_1:T | y_1:T)dx_1:T \]
and then maximizing \( Q(\theta, \theta_k) \) w.r.t. \( \theta \).

Problem: The E-step requires us to solve a smoothing problem, i.e., to compute an expectation under \( p_{\theta_k}(x_1:T | y_1:T) \).

SMC is used to approximate the smoothing pdf \( p_{\theta_k}(x_1:T | y_1:T) \).

Using EM and particle smoothing together

Algorithm 1 EM for identifying nonlinear dynamical systems
1. Initialise: Set \( k = 1 \) and choose an initial \( \theta_1 \).
2. While not converged do:
   (a) Expectation (E) step: Compute
   \[ Q(\theta, \theta_k) = \int \log p_\theta(x_1:T, y_1:T)p_{\theta_k}(x_1:T | y_1:T)dx_1:T \]
   using sequential Monte Carlo (particle smoother).
   (b) Maximization (M) step: Compute \( \theta_{k+1} = \arg \max_{\theta \in \Theta} Q(\theta, \theta_k) \)
   (c) \( k \leftarrow k + 1 \)

Example – blind Wiener identification

\[ x_{t+1} = \begin{pmatrix} A & B \end{pmatrix} \begin{pmatrix} x_t \\ u_t \end{pmatrix}, \quad u_t \sim N(0, Q), \]

\[ z_t = Cx_t, \quad y_t = h(z_t, \beta) + e_t, \quad e_t \sim N(0, R). \]

**Id. problem:** Find \( \mathcal{L}, \beta, r_1, \text{ and } r_2 \) based on \( \{y_{1,1:T}, y_{2,1:T}\} \).

---

Outline – Part 4

**Aim:** Show how SMC can be used to implement identification strategy 2 – Data augmentation.

1. Summary of Part 3
2. Identification strategy 2 – Data augmentation
3. Maximum likelihood via Expectation Maximization (EM)
4. **Sampling state trajectories using Markov kernels**
5. Bayesian identification using Gibbs sampling
6. Some of our current research activities (if there is time)
**Bayesian formulation** – model the unknown parameters as a random variable $\theta \sim p(\theta)$ and compute

$$p(\theta | y_{1:T}) = \frac{p(y_{1:T} | \theta)p(\theta)}{p(y_{1:T})} = \frac{p_0(y_{1:T})p(\theta)}{p(y_{1:T})}.$$ 

Gibbs sampling amounts to sequentially sampling from conditionals of the target distribution $p(\theta, x_{1:T} | y_{1:T})$.

A (blocked) example:

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

SMC is used to simulate from the smoothing pdf $p(x_{1:T} | y_{1:T})$.

**Problems and a solution**

Problems with this approach,

- Based on a PF $\Rightarrow$ approximate sample.
- Does not leave $p(x_{1:T} | \theta, y_{1:T})$ 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.


**Particle Gibbs (PG)**

The idea underlying **Particle Gibbs (PG)** is to make use of an SMC algorithm to construct a Markov kernel leaving the joint smoothing distribution $p(x_{1:T} | \theta, y_{1:T})$ invariant.

This Markov kernel is then used within a **standard** Gibbs sampler that operates on a non-standard space.

SMC is used to build an MCMC kernel with $p(x_{1:T} | \theta, y_{1:T})$ as its stationary distribution without introducing any systematic errors!
Three SMC algorithms

Three SMC algorithms leaving \( p(x_{1:T} | \theta, y_{1:T}) \) invariant:

1. Conditional particle filter (CPF)
   

2. CPF with backward simulation (CPF-BS)
   
   

3. CPF with ancestor sampling (CPF-AS)
   

CPF vs. CPF-AS – motivation

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.

CPF-AS – intuition

Let \( x_{1:T}' = (x_{1}', \ldots, x_{T}') \) be a fixed reference trajectory.

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

The CPF-AS causes us to degenerate to something that is very different from the reference trajectory, resulting in better mixing.
CPF-AS – algorithm

**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 \( x^N_1 = x^i_1 \).
   - (c) Set \( w^i_t = W_{\theta,1}(x^i_1) \).

2. **For** \( t = 2 \) **to** \( T \) **do**:
   - (a) Draw \( a^i_t \sim C(\{w^j_{t-1} \}_{j=1}^N) \), for \( i = 1, \ldots, N - 1 \).
   - (b) Draw \( x^i_t \sim r_{\theta,t}(x^i_t | x^i_{1:t-1}) \), for \( i = 1, \ldots, N - 1 \).
   - (c) Set \( x^N_t = x^i_t \).
   - (d) Draw \( a^N_t \) with \( P(a^N_t = i) \propto w^i_t \gamma_{\theta}(x^i_{1:t}, x_{1:t-1}) \).
   - (e) Set \( x^i_{1:t} = \{x^i_{1:t-1}, x^i_t\} \) and \( w^i_t = W_{\theta,t}(x^i_{1:t}) \).

The PGAS Markov kernel

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

- 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 \( p(x_{1:T} | \theta, y_{1:T}) \) invariant,

\[
p(dx^*_{1:T} | \theta, y_{1:T}) = \int P^N_{\theta}(x'_{1:T} dx^*_{1:T})p(dx'_{1:T} | \theta, y_{1:T})
\]
Toy example – stochastic volatility (I/II)

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, \( \hat{\theta} - E[\theta | y_{1:T}] \).

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.

Example – semiparametric Wiener model

"Parameters": \( \theta = \{A, B, Q, g(\cdot), r\} \).

Bayesian model specified by priors

- Conjugate priors for \( \Gamma = [A B] \), \( Q \) and \( r \),
  - \( p(\Gamma, Q) = \text{Matrix-normal inverse-Wishart} \)
  - \( p(r) = \text{inverse-Wishart} \)
- Gaussian process prior on \( g(\cdot) \),
\[ g(\cdot) \sim \mathcal{GP}(z, k(z, z')). \]

Inference using PGAS with \( N = 15 \) particles.
\( T = 1000 \) measurements. We ran 15 000 MCMC iterations and discarded 5 000 as burn-in.
Example – semiparametric Wiener model

Show movie

Bode diagram of the 4th-order linear system. Estimated mean (dashed black), true (solid black) and 99% credibility intervals (blue).


Inference in probabilistic graphical models

Constructing an artificial sequence of intermediate target distributions for an SMC sampler is a powerful (and quite possibly underutilized) idea.


Some of our current research activities

Joint work with (alphabetical order):
John Aston (University of Cambridge), Alexandre Bouchard-Côté (University of British Columbia), Pete Bunch (Quantcast), Adam M. Johansen (University of Warwick), Bonnie Kirkpatrick (University of Miami), Fredrik Lindsten (University of Cambridge), Christian A. Naesseth (Linköping University), Sumeetpal S. Singh (University of Cambridge), Arno Solin (Aalto University), Andreas Svensson (Uppsala University), and Simo Särkkä (Aalto University).

Gaussian Process nonlinear state space model

Consider the Gaussian Process SSM (GP-SSM):

\[ x_{t+1} = f(x_t) + w_t, \quad \text{s.t.} \quad f(x) \sim \mathcal{GP}(0, \kappa_{fg}(x, x')) \]

\[ y_t = g(x_t) + e_t, \quad \text{s.t.} \quad g(x) \sim \mathcal{GP}(0, \kappa_{fg}(x, x')) \]

The model functions \( f \) and \( g \) are assumed to be realizations from Gaussian process priors and \( w_t \sim \mathcal{N}(0, Q) \), \( e_t \sim \mathcal{N}(0, R) \).

We can now find the posterior distribution

\[ p(f, g, Q, R, \theta \mid y_{1:T}) \]

by making use of new MCMC algorithms.


Arcsine law for random variables. By making use of new MCMC algorithms.
Gaussian Process nonlinear state space model

This gives us a flexible nonparametric model where the GP prior on $f$ takes on the role of a regularizer. This provides a data-driven way of tuning the model complexity.

Toy example:

\[
x_{t+1} = -10 \frac{x_t}{1 + 3x_t^2} + \epsilon_t,
\]
\[
y_t = x_t + \epsilon_t.
\]

Near-degenerate/intractable state transition

Two model classes where state-of-the-art algorithms still struggle:

1. models where the state transition kernel is (nearly) degenerate, i.e. (nearly) concentrated on a low-dimensional manifold.

2. models where point-wise evaluation of the state transition density is intractable.

Main difficulty: The backward kernel will also be (nearly) degenerate or intractable, ruling out forward-backward methods.

New construction allowing us to replace simulation from the (problematic) backward kernel, with simulation from a joint density over a subset of future state variables.

The aim of this tutorial

The aim of this tutorial is to provide an introduction to the theory and application of sequential Monte Carlo (SMC) methods for nonlinear system identification (and state estimation) problems.

When it comes to SMC we will focus on the particle filter.

After this tutorial you should be able to derive your own SMC based algorithms allowing you to learn nonlinear dynamical models and to estimate the states in these models.
Conclusions

<table>
<thead>
<tr>
<th>Marginalization</th>
<th>Data augmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML Direct optimization</td>
<td>Expectation Maximization</td>
</tr>
<tr>
<td>Bayesian Metropolis Hastings</td>
<td>Gibbs sampling</td>
</tr>
</tbody>
</table>

SMC can be used to realize these strategies for nonlinear SSMs.

Key capabilities of SMC – approximate solutions to
1. Compute the likelihood and its derivatives.
2. Solve state smoothing problems, e.g. compute $p(x_{1:T} \mid y_{1:T})$.
3. Simulate from the smoothing pdf, $\tilde{x}_{1:T} \sim p(x_{1:T} \mid y_{1:T})$.

Fast moving research area offering lots of opportunities!